

Doppelausgabe
März/April 86
3. Jahrgang
2
Computronic
software
für Heimcomputer
ÖS 55 Sfr 6,50 DM 6,50
Hits aus dem Inhalt
C=64
C=16
VIC 20
ATARI
Schneider
TI-99/4A
apple
sinclair Spectrum
1986
Riesen-Chance
für alle
Software-Autoren
Im Heft
Checksummer
für
Commodore 64
Schneider CPC 464
MSX-Systeme
VC-20
Spaß mit dem Computer
Superspiele
zum Eingeben!


SONDER-
Neu!
Ausgabe
Computronic
software für Heimcomputer
über
28
... erhältlich überall im Zeitschriftenhandel!
Superspiele
C=64
C=16
Schneider
CPC464
ATARI
MSX

So werden Fehlerquellen vermieden!

COMPUTRONIC-CHECKSUMMER

Achtung!... Achtung!... Computronic-Checksummer jetzt auch für den VC-20!!

Nach dem großen Zuspruch unseres Checksummers für die Systeme C-64 und Schneider, hat unser Spezialist Frank Brall jetzt auch eine Version für den VC-20 entwickelt. Dadurch sollen die Vorteile unseres Prüfsummenverfahrens auch den VC-Freunden zugute kommen. Tippfehler und unklare Steuerzeichen gehören damit der Vergangenheit an.
Für unsere neu hinzugekommenen Leser drucken wir an dieser Stelle noch einmal die Bedienungsanleitung für die gesamten Checksummer ab. Nach C-64, CPC-464 und VC-20 arbeitet jetzt unserer Spezialist an einem Prüfsummenverfahren für den C-16/116. Wir hoffen, diesen Checksummer bald vorstellen zu können.

Eingabe-Hinweise für Commodore-C-64/ VC-20-Benutzer!

Um die Eingabe unserer Listings zu erleichtern, werden zukünftig alle C-64/VC-20-Listings ohne Steuerzeichen abgedruckt.
Alle Steuerzeichen werden in unseren LISTINGS durch Klartext, welcher die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden sie in jedem Tronic-Magazin.

Das Beispiel auf Seite 7 demonstriert die Arbeitsweise unseres Systems!

Wie in dem Beispiel zu erkennen ist, wird beispielsweise das Herz-Symbol durch das Wort CLEAR ersetzt. Dies bedeutet, Sie müssen die Taste CLEAR drücken, um das Herz-Symbol zu erzeugen. Sollten Sie einmal nicht wissen, welche Taste gemeint ist, so hilft Ihnen schnell unsere Tastentabelle, welche in jedem Magazin abgedruckt ist.
Um im Listing Tastenbezeichnungen von Grafik oder normalen Texten zu unterscheiden, werden alle Tastenkennzeichnungen in geschweifte Klammern gesetzt. Diese dürfen selbstverständlich nicht eingegeben werden. Auch Leerzeichen innerhalb geschweifter Klammern dienen nur zur Trennung einzelner Tastenfunktionen und dürfen ebenfalls nicht eingegeben werden. Um die Tastenfunktionen noch besser hervorzuheben, werden diese in unterstrichener Kursivschrift (Schrägschrift) abgedruckt.

Steht hinter einer Tastenfunktion eine Zahl, welche ebenfalls unterstrichen ist, so bedeutet dies, daß die letzte Tastenfunktion mehrmals betätigt werden muß. Die in unserem Beispiel abgedruckte Funktion RIGHT3 bedeutet, daß die Taste RIGHT (Cursor rechts) 3 mal hintereinander betätigt werden muß. Auch einzelne oder auch mehrere Leerzeichen innerhalb von Anführungszeichen werden auf diese Art gekennzeichnet. Das bislang übliche Abzählen einzelner Zeichen entfällt somit völlig. Alle Zeichen außerhalb von den geschweiften Klammern werden normal abgedruckt und auch eingegeben.
Auf den ersten Blick hört sich das sicher etwas kompliziert an, ist jedoch in der Praxis ganz einfach. Wenn man sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt hat, wird man den großen Vorteil dieser Schreibweise erkennen.



Schon gehört? Bei COMPUTRONIC gibt's den Checksummer jetzt auch für VC-20!

Checksummer C Version 1.0 für Commodore 64

Checksummer v1.0 ist ein Prüfsummen-Programm, das die Eingabe von Programm-Listings zum wahren Vergnügen macht. Tippfehler werden schon während des Eingebens der einzelnen Programmzeilen erkannt. Dieses System, zusammen mit einem neuen Druckverfahren, gewährleisten, daß unsere Listings zu 99.9% fehlerfrei abgedruckt werden.

Arbeitsweise und Aufbau unseres Checksummers:
Unser Checksummer besteht aus einem kleinem Maschinenprogramm, welches als Basic-Loader abgedruckt ist.
Tippen Sie diesen Loader ein und speichern Ihn auf Kassette oder Diskette ab; Sie können Ihn zukünftig immer wieder benutzen. Der Start erfolgt durch den Befehl „RUN“. Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC ..“. Der Checksummer ist nun aktiv. Nun kann man ein beliebiges Tronic-Listing eingeben. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem Schirm eine Prüfzahl. Vergleichen Sie diese mit der Zahl, welche im Heft hinter diese Zeile abgedruckt ist. Stimmen die Zahlen überein, so ist die Zeile richtig eingegeben; ansonsten muß sich noch ein Fehler in Ihre Eingabe eingeschlichen haben, und Sie müssen die Zeile korrigieren. Auf diese Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.
Interressant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen be-

Lesen Sie weiter auf Seite 6!

Kurz belichtet

C-64

INTERCEPTOR BASE

Ronald Mayer, Software-Champion des Jahres 1985, hat wieder einmal zugeschlagen! Mit **Interceptor Base** ist ihm einmal mehr ein wahres Meisterwerk gelungen. Schon beim ersten Blick auf das Spiel fallen seine außergewöhnlich gute Grafik und die enorm hohe Spielgeschwindigkeit auf. Interessant auch die Spielidee: Was anfangs nach einer



alltäglichen Routineaufgabe aussieht, entwickelt sich binnen kürzester Zeit zu einem Kampf auf Leben und Tod! Denn kaum sind Sie mit Ihrem Düsenjäger zu einem Patrouillenflug gestartet, da werden Sie von feindlichen Abfangjägern angegriffen. Es entbrennt ein „Fight", den Sie nur dann bestehen können, wenn Sie all Ihr Geschick und Können spielen lassen. **(bez.)**

Seite 8



C-64

SCHOTTER

Bei diesem Spiel geht es um bare „Kohle"! Und zwar um die „Penunze" des **TRONIC-Verlages** – das allein macht **SCHOTTER** doch sicher für Sie interessant, oder? Doch nun genug der Vorrede, wir wollen Sie nicht länger auf die Folter spannen. Die Handlung von **SCHOTTER** ist im wahrsten Sinn des Wortes originell: Nachdem



Sie im Schweiße Ihres Angesichts ein Software-Programm fertiggestellt haben, müssen Sie nur noch in Wehretal (Sitz des **TRONIC-Verlages)** Ihr Autorenhonorar eintreiben. Um aber in den Besitz des sauerverdienten „Schotters" zu kommen, müssen Sie zunächst einmal einige Hürden überspringen! **(bez.)**

Seite 15

Lieber Leser!

Auf vielfachen Wunsch hat sich die Redaktion entschlossen, die in COMPUTRONIC veröffentlichen Computer-Spiele unserer Software-Autoren in verkürzter Form zu Beginn des Heftes vorzustellen. Im Innenteil der Ausgabe erhalten Sie dann weitere ausführliche Informationen. Leider war es uns diesmal noch nicht möglich, zu jedem Spiel ein Bildschirmfoto abzudrucken; wir werden uns aber bemühen, in Zukunft auch dies für Sie zu realisieren.

Wir hoffen, daß diese kleine Service-Leistung auf Ihre Zustimmung stößt und würden uns freuen, wenn Sie uns Ihre Meinung hierzu mitteilen. Schreiben Sie uns doch einfach mal.

Die Redaktion

VC-20

MINER
DER FASSADENSTREICHER

Mit **Miner, der Fassadenstreicher** können wir Ihnen ein „Gerüstspiel" der Spitzenklasse präsentieren! Miner, der Fassadenstreicher muß zu seinem Leidwesen feststellen, daß Handwerk nicht nur „goldenen Boden" hat, sondern mitunter auch sehr gefährlich werden kann. Während Miner nichtsahnend und fröhlich pfeifend seiner

TOLLES SPIEL- SUPER GRAFIK!

Arbeit nachgeht, haben es einige finstere Zeitgenossen (vielleicht eine Konkurrenzfirma?) darauf abgesehen, ihn in schwindelnder Höhe von seinem wackligen Gerüst zu stoßen. Da gerät Miner unversehens ja in einen ganz schönen Schlamassel! Aber als „alter Hase" auf dem Gerüst wird es ihm schon gelingen, seinen Gegnern zu entgehen, oder...? **(bez.)**

Seite 20



VC-20

INKA-SCHATZ

Begleiten Sie den großen spanischen Eroberer Hernando Cortez auf seiner abenteuerlichen Schatzsuche im Land der Inkas! Aber seien Sie vorsichtig: im heißen Mexiko lauern tausend Gefahren auf Sie! Nicht nur die Inkas machen Ihnen das Leben schwer, sondern auch allerhand Getier hat es auf Sie abgesehen. Wenn Sie also den **Inka-Schatz** bergen wollen, so müssen Sie sich erst einmal mit wilden Hunden, angriffslustigen Adlern und giftigen Schlangen auseinandersetzen. Dieses Spiel wird Ihnen Ihr ganzes Können abverlangen! **(bez.)**

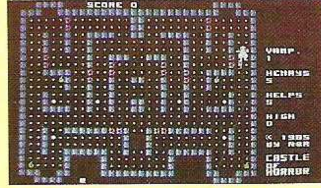
Seite 24



C-16

HORROR CASTLE

Ein Spiel für Leute, die's schaurig mögen. Ihr vermögender Urgroßvater hat Ihnen ein altes Schloß vermacht (so weit ist das ja noch ganz angenehm). Doch das Gemäuer liegt im mystischen schottischen Hochland, und da wird's einem nun allmählich

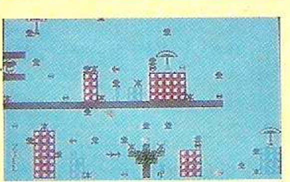


doch schon etwas „mulmig". Doch trotz aller Bedenken und Warnungen machen Sie sich auf den Weg, Ihr Erbe anzutreten. Aber erst einmal in der alten Burg angekommen, sollen Sie Ihren Entschluß sehr schnell bereuen. Denn **Horror Castle** macht seinem Namen alle Ehre! **(bez.)**

Seite 27

C-16

FANTASY COUNTRY



Fantasy Country entführt Sie in eine bezaubernde, märchenhafte Welt voller Abenteuer. Will man etwas über die Handlung

Kurz belichtet

des Spiels sagen, so fällt einem spontan die Geschichte um „Sterntaler“ ein, denn genau dies ist Ihre Aufgabe: Sie müssen im **Fantasy Country** möglichst viele Sterne einsammeln, und diese sind gleichzeitig das einzige, was Sie im „Land der Phantasie“ berühren dürfen. Das ist wahrlich nicht einfach, denn es tummelt sich so einiges auf dem Bildschirm! **(bez.)**

Seite 35

SPECTRUM

OTTO SCHWEINSOHR

Nicht umsonst wurde **Otto Schweinsohr** zu einem der Top-Spiele erkoren. Wie schon der Titel vermuten läßt, liegt diesem Spiel eine überaus originelle Idee zugrunde. Otto Schweinsohr führt einen unerbittlichen



Kampf gegen den fürchterlichen Clan der Raubkopierer. Doch nicht nur diese haben es auf Otto abgesehen, sondern auch der interstellare Geheimdienst verfolgt ihn wegen eines Vergehens gegen zahlreiche Vorschriften. **(bez.)**

Seite 39

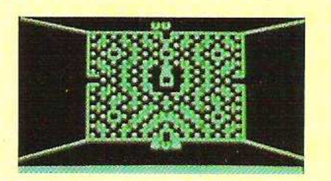


APPLE

PAC Boy

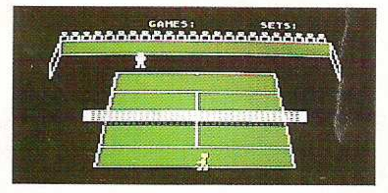
Immer wenn er Pillen nahm, seine große Stunde kam! Helfen Sie **Pac Boy** bei seiner Jagd nach den kraftspendenden Pillen. Er braucht diese „Super-Dragees“ so dringend wie andere das täglich' Brot! Pac Boy's Reise durch ein finsteres Labyrinth kostet den kleinen Burschen eine Menge Energien. Um neue Kräfte sammeln zu können (die im Kampf gegen das gefährliche Pac-Monster auch dringend benötigt werden), gibt es nur ein Mittel: (fr)essen, was das Zeug hält! Die ansprechende Grafik macht das Spiel wirklich sehenswert – am besten, Sie überzeugen sich selbst! **(bez.)**

Seite 48



ATARI

TOP TENNIS



Im Zuge des „Becker-Booms“ kann man fast nicht mehr anders – man muß sich einfach für Tennis interessieren! Mit **Top Tennis** haben wir da für Sie genau das richtige Spiel „an Land gezogen“. Schlüpfen Sie in die Haut des „Bundes-Boris“, und fegen Sie Ihre Gegner reihenweise vom Platz. Aber bis Sie es so weit gebracht haben, müssen Sie schon hart trainieren, denn **Top Tennis** stellt außerordentlich hohe Anforderungen an den Spieler! **(bez.)**

Seite 52



ATARI

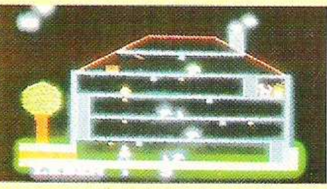
PANZERSCHLACHT

Panzerschlacht bringt Ihnen die Action ins Haus! Steuern Sie Ihren Tank durch die gegnerischen Linien, aber lassen Sie alle Vorsicht walten, denn der Feind ist Ihnen zahlenmäßig weit überlegen! 15mal stärker sind die angreifenden Truppenverbände – da brauchen Sie schon Ihr ganzes Geschick, aber auch ein wenig Glück, um in diesem aussichtslos erscheinenden Kampf bestehen zu können. **Panzerschlacht** beeindruckt durch seine enorme Spielgeschwindigkeit – Sie werden staunen (aber staunen Sie nicht zu lange, der Feind macht sonst kurzen Prozess mit Ihnen)! **(bez.)**

Seite 61

CPC-464

MIDNIGHT



Seien Sie vorsichtig bei **Midnight**! In diesem spannenden Spiel haben Sie es mit schaurigen Gespenstern zu tun, die ein halb verfallenes, verrufenes Haus heimgesucht haben. Um Mitternacht kommt es dann zu der letzten, alles entscheidenden Auseinandersetzung mit den Geistern. Ob Sie diesen Ausgeburten der Hölle wohl gewachsen sind?! **(bez.)**

Seite 65

CPC-464

HORROR CAVES

Gänsehaut gefällig? Bitte – wenn Sie es nicht anders wollen! Dann steigen Sie mal ein in unseren „Horror-Express“, und folgen Sie uns in die Geister-Höhlen der Pyrenäen. Wir versprechen Ihnen das absolute Grauen. Na, nun ist Ihnen wohl doch nicht mehr ganz wohl zumute? Aber jetzt ist es zu spät, nun gibt es kein Zurück mehr. Der Zug rollt, und er wird erst wieder in den **Horror Caves** zum Stehen kommen. **(bez.)**

Seite 71

TI-99

THAI BOXING

Sind Sie ein Anhänger der Selbstverteidigung? Mögen Sie Kampfsportarten? Interessieren Sie sich für die asiatische Form des Kämpfes? Wenn Sie diese Fragen mit „ja“ beantworten, dann ist **Thai-Boxing“** für Sie genau das Richtige! So wie der Thai-Boxer eiserne Nerven und absolute Körperbeherrschung aufbringen muß, benötigen Sie jede Menge Geschick und Ausdauer. Um es kurz zu sagen: **Thai-Boxing** ist wahrlich ein echter „Klopper“! **(bez.)**

Seite 75

Der weitere INHALT

nutzt werden können, ohne die Checksumme zu verändern. Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben. Sie können Ihr Programm auch starten, denn der Checksummer und Ihr Programm beeinflussen sich nicht gegenseitig. Wollen Sie den Checksummer abschalten, so geben Sie einfach „POKE 1,55" ein oder betätigen die Tasten-Kombination „RUNSTOP und RESTORE". Aktivieren können Sie den Checksummer jederzeit (auch nach RESET) mit „POKE 1,53".

Wer den Checksummer nicht eingeben möchte, kann diesen auch unter der folgenden Bestellnummer beziehen:

Bestell-Nr. CV10K / Kassette 10 DM
CV10D / Diskette 15 DM

```
C64 LISTING MIT CHECKSUMMEN (C V1.0)

0 REM ***** TRONIC CHECKSUMMER *****           <116>
1 REM VERSION C 1.0 BEI FRANK BRALL            <10>
2 REM ------------------------------           <148>
3 FOR I=832 TO 1008:READ A:S=S+A               <68>
4 POKE I,A:NEXT I:IF S<>16397 THEN PRINT
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE
}!{SPACE}(10-19)":STOP                         <32>
5 SYS 832:S=0:FOR I=58464 TO 58603:READ
A:S=S+A                                        <21>
6 POKE I,A:NEXT I:IF S<>18919 THEN PRINT
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE
}!{SPACE}(29-36)":STOP                         <219>
7 S=0:FOR I=48979 TO 49005:READ A:S=S+A        <33>
8 POKE I,A:NEXT I:IF S<>2888 THEN PRINT"
FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE}
!{SPACE}(46-47)":STOP                          <23>
9 PRINT"{DOWN2}AKTIVIEREN{SPACE}:{SPACE}
POKE{SPACE}1,53{SPACE18}AUSSCHALTEN:{SPA
CE}POKE{SPACE}1,55":NEW                        <26>
10 DATA169,0,133,254,162,1,189,137,3,133
,255,160,0,177,254,145,254,136                 <176>
11 DATA208,249,230,255,165,255,221,139,3
,208,238,202,16,230,169,96,141,49              <147>
12 DATA165,169,228,141,50,165,169,53,133
,1,169,141,133,254,162,3,134,255               <225>
13 DATA160,0,177,254,240,7,32,202,241,20
0,76,120,3,169,2,141,32,208                    <140>
14 DATA96,160,224,192,0,147,17,32,32,35,
35,35,35,32,84,82,79,78                        <158>
15 DATA73,67,45,86,69,82,76,65,71,32,67,
72,69,67,75,83,85,77                           <129>
16 DATA77,69,82,32,35,35,35,35,139,13,13
,32,32,32,32,32,32,32                          <147>
17 DATA32,32,32,32,32,86,69,82,83,73,79,
78,32,67,32,49,46,48                           <79>
18 DATA13,13,32,32,32,32,32,32,40,67,41,
32,70,82,65,78,75,32                           <94>
19 DATA66,82,65,76,76,32,40,49,48,46,56,
53,41,13,0                                     <12>
29 DATA169,0,141,107,191,141,108,191,141
,109,191,160,2,24,177,95,200,113               <7>
30 DATA95,141,109,191,160,3,200,177,95,2-
40,44,201,34,208,10,173,107,191                <91>
31 DATA73,1,141,107,191,177,95,174,107,1
91,208,4,201,32,240,228,238,108                <128>
32 DATA191,174,108,191,24,177,95,109,109
,191,141,109,191,202,208,244,76,120            <244>
33 DATA228,56,76,183,228,72,32,201,255,1
70,104,144,1,138,96,32,240,255                 <101>
34 DATA142,105,191,140,106,191,162,0,160
,0,24,32,240,255,169,91,32,12                  <89>
35 DATA225,169,0,174,109,191,32,205,189,
169,93,76,83,191,173,33,208,145                <239>
36 DATA243,96,105,2,164,145,200,208,4,19
7,161,208,247,96                               <202>
46 DATA32,12,225,32,63,171,32,63,171,172
,106,191,174,105,191,24,32,240                 <213>
47 DATA255,76,128,164,24,0,0,5,200             <65>
  ENDE DES LISTINGS
```

Checksummer für COMMODORE 64

Checksummer VC 1.0 für VC-20

Ebenso wie der C-64-Checksummer besteht auch das VC-20-Programm aus einer kleinen Maschinenroutine. Aus Speicherplatzgründen konnte jedoch dieses Programm nicht „unter" dem ROM abgelegt werden, sondern hier mußte der bekannte Kassettenpuffer herhalten. Dies hat den Vorteil, daß kein Basic-Speicher verloren geht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchführbar sind, da diese den Kassettenpuffer benötigen. Um die Kassettenoperationen wieder zu ermöglichen, muß der Checksummer durch den Befehl „SYS 58459", sowie durch das Betätigen der Tasten STOP/RESTORE, abgeschaltet werden. Aktiviert wird der Checksummer mit SYS 828. Die Funktion und Arbeitsweise stimmt mit dem C-64-Checksummer überein und kann dort nachgelesen werden.

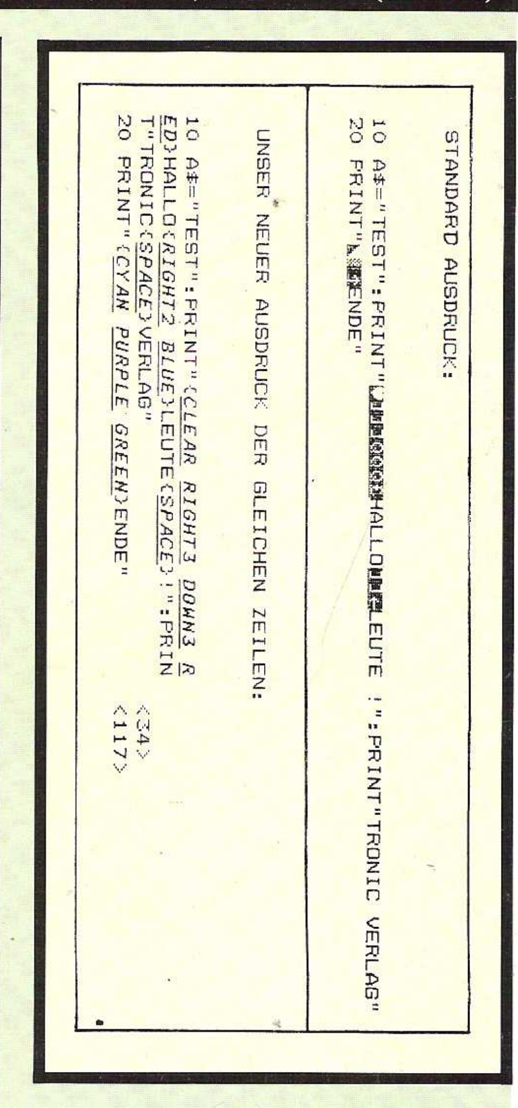
```
1 REM *****************************            <176>
2 REM      CHECKSUMMER VC 1.0                  <126>
3 REM        VC 20 VERSION                     <8>
4 REM     COPYRIGHT  FRANK BRALL               <126>
5 REM   T R O N I C - V E R L A G              <228>
6 REM *****************************            <181>
7 REM                                          <150>
10 PRINT"{CLEAR DOWN SPACE}CHECKSUMMER{S
PACE}VC{SPACE}1.0{SPACE2}"                     <216>
```

```
20 PRINT "{DOWN}COPYRIGHT{SPACE}FRANK{SP
ACE}BRALL"                                  <221>
30 PRINT "{DOWN}-----{SPACE}03.01.86{SPA
CE}-----"                                   <152>
35 PRINT "{DOWN SPACE4}TRONIC-VERLAG"       <123>
60 DA=0:FORI= 828 TO 1020                   <158>
61 READ DA:SU=SU+DA                         <163>
62 POKE I,DA                                <64>
63 NEXT I                                   <83>
65 IF SU<>22919 THEN PRINT"{DOWN}FEHLER{
SPACE}IN{SPACE}DATA-ZEILEN{SPACE}!":END     <14>
70 PRINT"{DOWN}"                            <222>
71 PRINT"AKTIVIEREN{SPACE}=SYS{SPACE}828
"                                           <9>
72 PRINT"ABSCHALTEN{SPACE}=SYS{SPACE}584
59"                                         <74>
73 PRINT"VOR{SPACE}BETRIEB{SPACE}DER{SPA
CE}DATA-"                                   <40>
74 PRINT"SETTE{SPACE}DEN{SPACE}CHECKSUMM
ER"                                         <38>
75 PRINT"ABSCHALTEN{SPACE}UND{SPACE}DIE"    <80>
76 PRINT"TASTEN{SPACE}STOP/RESTORE"         <198>
77 PRINT"BETAETIGEN{SPACE}!"                <19>
80 POKE 816,116:POKE 818,116:POKE 817,19
6:POKE 819,196:REM SAVE/LOAD AUS            <151>
90 SYS 828:PRINT"{DOWN}CHECK{SPACE}1.0{S
PACE}IST{SPACE}AKTIV{SPACE}!"               <99>
100 DATA 169,71,141,2,3,169,3,141,3,3,96
,32,105,3,134,122,132                       <85>
101 DATA 123,32,115,0,170,240,243,162,25
5,134,58,144,8,134,255,32,121               <100>
102 DATA 197,76,225,199,162,1,134,255,76
,156,196,166,255,224,1,240,3                <161>
103 DATA 76,96,197,169,0,141,248,3,141,2
49,3,141,250,3,160,2,24                     <211>
104 DATA 177,95,200,113,95,141,250,3,160
,3,200,177,95,240,44,201,34                 <143>
105 DATA 208,10,173,248,3,73,1,141,248,3
,177,95,174,248,3,208,4                     <94>
106 DATA 201,32,240,228,238,249,3,174,24
9,3,24,177,95,109,250,3,141                 <31>
107 DATA 250,3,202,208,244,76,138,3,56,3
2,240,255,142,246,3,140,247                 <203>
108 DATA 3,162,0,160,0,24,32,240,255,169
,91,32,9,225,169,0,174                      <232>
109 DATA 250,3,32,205,221,169,93,32,9,22
5,32,63,203,32,63,203,172                   <21>
110 DATA 247,3,174,246,3,24,32,240,255,1
62,0,134,255,76,111,3,0                     <121>
111 DATA 0,0,0,0,32,0                       <118>
```

Der VC-20-Checksummer kommt frei Haus unter folgenden Bestellnummern: VV10K (Kassette) oder VV10D (Diskette)

Tastenbezeichnungen für Steuercodes (TRONIC-NORM)

DOWN	CURSOR UNTEN	TASTE NEBEN RECHTEM SHIFT
UP	CURSOR HOCH	SHIFT-TASTE & TASTE NEBEN RECHTEM SHIFT
CLR	CLEAR SCHIRM	SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN
INST	EINFUEGEN	SHIFT-TASTE & TASTE GANZ RECHTS OBEN
HOME	CURSOR IN ECKE	2. TASTE VON GANZ RECHTS OBEN
DEL	DELETE	TASTE GANZ RECHTS OBEN
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS UNTEN
LEFT	CURSOR LINKS	SHIFT-TASTE & TASTE UNTEN RECHTS
SPACE	LEERZEICHEN	LEERTASTE (GROESSTE TASTE)
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8
BLACK	SCHWARZ	CONTROL-TASTE & 1
WHITE	WEISS	CONTROL-TASTE & 2
RED	ROT	CONTROL-TASTE & 3
CYAN	TUERKIS	CONTROL-TASTE & 4
PURPLE	PURPUR	CONTROL-TASTE & 5
GREEN	GRUEN	CONTROL-TASTE & 6
BLUE	BLAU	CONTROL-TASTE & 7
YELLOW	GELB	CONTROL-TASTE & 8
RVSON	INVERSE EIN	CONTROL-TASTE & 9
RVSOFF	INVERSE AUS	CONTROL-TASTE & 0
ORANGE	ORANGE	COMMODORE-TASTE & 1
BROWN	BRAUN	COMMODORE-TASTE & 2
LIG.RED	HELLROT	COMMODORE-TASTE & 3
DGREY	DUNKELGRAU	COMMODORE-TASTE & 4
MGREY	MITTELGRAU	COMMODORE-TASTE & 5
LIG.GREEN	HELLGRUEN	COMMODORE-TASTE & 6
LIG.BLUE	HELLBLAU	COMMODORE-TASTE & 7
HGREY	HELLGRAU	COMMODORE-TASTE & 8
CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN MIT DEM NACHFOLGENDEN ZEICHEN. (Z.B. CTRLA ENTSPRICHT CTRL & A)



```
STANDARD AUSDRUCK:

10 A$="TEST":PRINT"[graphics]HALLO[graphics]LEUTE !":PRINT"TRONIC VERLAG"
20 PRINT"[graphics]ENDE"

UNSER NEUER AUSDRUCK DER GLEICHEN ZEILEN:

10 A$="TEST":PRINT"{CLEAR RIGHT3 DOWN3 R
ED}HALLO{RIGHT2 BLUE}LEUTE{SPACE}!":PRIN
T"TRONIC{SPACE}VERLAG"                     <34>
20 PRINT"{CYAN PURPLE GREEN}ENDE"          <117>
```

Fortsetzung S. 64



Interceptor-Base

Unser Software-Champion '85 Ronald Mayer hat mit diesem Super-Game seinem Titel wieder zur Ehre gereicht. Interceptor-Base beeindruckt durch seine hervorragende Grafik und seine hohe Spielgeschwindigkeit.
Eigentlich befinden Sie sich mit Ihrem Düsenjäger auf einem ganz alltäglichen Routineflug. Doch kaum gestartet, geraten Sie innerhalb kürzester Zeit in die größte Bedrängnis. Ihr Jet wird von feindlichen Abfangjägern angegriffen, die gegnerische Flugabwehr versucht Sie vom Himmel zu holen und zu allem Überfluß müssen Sie Ihren Treibstoffvorrat immer wieder durch Zerstören fremder Öllager auffüllen. Je länger Sie sich am umkämpften Himmel halten, desto gefährlicher wird es für Sie. Dafür sorgen die feindlichen Ufos und Helicopter, die nach Erreichen einer bestimmten Punktzahl plötzlich am Firmament auftauchen. Doch nicht aufgeben, irgendwann hat man sich bestimmt wieder die Luftherrschaft zurückerobert. (S.G.)



Spielbeschreibung:

Ziel des Spiels: Soviele Punkte wie möglich zu sammeln und darauf zu achten, daß der Treibstoff nicht ausgeht.

Levels: Der Grundlevel wird vom Spieler eingestellt (Anfänger, Standard, Experte). Nachdem man eine gewisse Strecke zurückgelegt hat, erhöht sich der Spritverbrauch und die Art der Gegner.

Die Gegner:

JET1: Zählt 100 Punkte. Feuert mit Raketen. Fliegt eine gerade Linie. Langsam.

JET2: Zählt 200 Punkte. Feuert mit Raketen. Starten, fliegt in die Höhe Ihres Abfangjägers. Langsam.

Destroyer: Zählt 300 Punkte. Feuert mit Raketen. Fliegt eine gerade Linie. Schnell.

Helicopter: Zählt 500 Punkte. Feuert mit Raketen. Versucht immer auf Ihrer Höhe zu bleiben. Sehr angriffslustig!!!

Bonusufo: Zählt 1000 Punkte Bonus und feuert ebenfalls mit Raketen. Es erscheint sehr selten und macht sich daher durch einen Signalton bemerkbar.

Kanonen: Sind am Boden stationiert und feuern mit Raketen.

Treibstoff: Durch Zerstören der mit „OIL" beschrifteten Gebäude wird der Treibstoffvorrat erhöht.

Bewaffnung: Unbegrenzter Vorrat an Luft-Raketen und Bomben. Alle sich in der Luft befindlichen Objekte können nur mit Raketen zerstört werden, alle sich am Boden befindlichen nur mit Bomben.

Bonus: Hat man über 10000 bzw. über 40000 Punkte erreicht bekommt man ein zusätzliches Flugzeug.

Armaturen: SCORE: Anzahl der Punkte
FUEL: Tankanzeige
PILOT: Anzahl Ihrer Flugzeuge

Spielende: Das Spiel ist beendet, wenn Ihre gesamten Flugzeuge zerstört sind.

Abtippen, Laden und Starten des Programms

Das Abtippen der Programme:

1. Screenmaker, Grafikmaker, Mcodemaker und Interceptor abtippen und abspeichern.
2. Screenmaker laden und starten. Es werden nun die Bildschirmcodes für das Spielscreen ab Adresse 4096 erzeugt. Nach diesem Vorgang erscheint die Meldung „Please load Grafikmaker".
3. Grafikmaker laden und starten. Dieses Programm erzeugt die Sprites und den neuen Zeichensatz. Weiter wird ein kleines Startprogramm am Basic-Start erzeugt.
4. Mcodemaker laden und starten. Nun wird der Maschinencode erzeugt.
5. Interceptor laden. Dies ist das Hauptprogramm. Das Programm noch nicht !!! starten !!!
6. Nach einer fehlerlosen Ausführung der Punkte 1 – 5 befindet sich jetzt das komplette Programm im Speicher. Nun geben Sie bitte "Poke 44,8" + <Return> ein. Mit Save "Interceptor-Base",8(,1) können Sie nun das gesamte Programm auf Diskette (Kassette) abspeichern.

Diese Vorgänge (1 – 6) sind natürlich nur beim ersten Erstellen des Programms notwendig.

Das Laden und Starten des Programms:

1. Das Programm Interceptor-Base laden und wie üblch mit "Run" starten.
2. Joystick mit Port 2 verbinden und los geht's!!!

An diesem Spiel werden selbst die „abgebrühtesten" Computer-Freaks ihre Freude haben!

```
   C64 LISTING MIT CHECKSUMMEN (C V1.0)

100 REM********************************   <211>
101 REM* INTERCEPTORBASE-SCREENMAKER  *   <174>
102 REM*       (C)1985 BY MR-SOFT     *   <8>
103 REM*   R O N A L D    M A Y E R   *   <128>
104 REM*  A-4062  THENING 24/AUSTRIA  *   <168>
105 REM********************************   <216>
106 DATA"{SPACE}=>?->♠{SPACE9}"   <116>
107 DATA"{SPACE}|‾‾‾_|{SPACE9}"   <217>
108 DATA"{SPACE8}=>♠?->♠{SPACE}"   <48>
109 DATA"{SPACE8}|‾‾‾‾_|{SPACE}"   <179>
110 DATA"{SPACE16}"   <187>
111 DATA"=>♠{SPACE2}=‾‾>♠{SPACE6}"   <78>
112 DATA"|_|{SPACE2}|‾‾‾‾{SPACE2}=>♠{SPA
CE}"   <179>
113 DATA"{SPACE12}|‾‾{SPACE}"   <113>
114 DATA"{SPACE9}=>♠{SPACE4}"   <111>
115 DATA"=>♠{SPACE6}|_|{SPACE4}"   <220>
116 DATA"|‾‾{SPACE2}=>?->♠{SPACE2}=>♠"   <49>
117 DATA"{SPACE5}|‾‾‾_|{SPACE2}‾‾‾"   <253>
118 DATA3,1,3,2,1,3,1,1,3,2,1,3,1,3,2,2   <235>
119 POKE56,4096/256:CLR:FORI=1TO3:FORJ=0
TO3:READW$(I,J):NEXTJ,I   <57>
120 FORI=0TO15:READW   <35>
121 FORJ=0TO3   <202>
122 FORK=1TO16:POKE4096+256*J+I*16+K-1,A
SC(MID$(W$(W,J),K,1))ANDNOT128:NEXTK,J,I   <51>
123 DATA"{SPACE5}LL{SPACE5}L{SPACE3}"   <72>
124 DATA"{SPACE2}|\ ∪/\{SPACE}|\ ∪\{SPAC
E2}"   <96>
125 DATA"{SPACE}●┌┬┬┬┬┬┬┬┬┬┬┐{SPACE}-{SPA
CE}"   <172>
126 DATA"{SPACE}|\ \{SPACE2}-{SPACE}|\ \{S
PACE}:;<{SPACE}"   <1>
127 DATA"{SPACE16}"   <204>
128 DATA"{SPACE}♥////////////_./{SPACE}"   <120>
129 DATA"{SPACE}:;;;;;;<-●┌┬┬┬┬"   <114>
130 DATA"{SPACE3}-{SPACE6}-{SPACE}-{SPAC
E3}"   <137>
131 DATA"{SPACE4}L{SPACE11}"   <132>
132 DATA"{SPACE}:;;/\{SPACE}./././{SPA
CE}"   <151>
133 DATA"┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬"   <170>
134 DATA"{SPACE2}-{SPACE2}-{SPACE2}-{SPA
CE2}-♥/_{SPACE}"   <240>
135 DATA"{SPACE2}L{SPACE2}┘{SPACE2}L{SPA
CE7}"   <236>
136 DATA"{SPACE}|\ \|\ \|\ \{SPACE}♥////"   <9>
137 DATA"┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬"   <174>
138 DATA"{SPACE}:;<{SPACE2}-{SPACE2}:<:<
:<{SPACE}"   <19>
139 DATA"{SPACE16}"   <216>
140 DATA"//////////////_-{SPACE2}-♥"   <6>
141 DATA"┌┬┬┬┐●┐:;;<●┐●┌┬"   <136>
142 DATA"{SPACE2}-{SPACE}|\ \{SPACE}-{SPA
CE3}./{SPACE2}"   <71>
143 DATA"{SPACE3}L{SPACE12}"   <228>
144 DATA"////_./././././-"   <248>
145 DATA"┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┐"   <200>
146 DATA":;;<./././././-{SPACE}"   <43>
147 DATA"{SPACE16}"   <224>
148 DATA"|\ \{SPACE3}L{SPACE2}♥///////"   <117>
149 DATA"{SPACE3}|\ ∪\●┌┬┬┬┬┬┬┬"   <232>
150 DATA"./{SPACE}-{SPACE}./{SPACE}-{SPA
CE}:;;;<{SPACE}"   <10>
151 DATA1,2,3,4,5,6,1,7,6,2,4,5,6,1,7,6   <208>
152 FORI=1TO7:FORJ=0TO3:READW$(I,J):NEXT
J,I   <125>
153 FORI=0TO15:READW   <68>
154 FORJ=0TO3   <235>
155 FORK=1TO16:POKE5120+256*J+I*16+K-1,A
SC(MID$(W$(W,J),K,1))ANDNOT128:NEXTK,J,I   <199>
156 POKE53281,0:PRINT"{CLEAR YELLOW}PLEA
SE{SPACE}LOAD{SPACE}GRAFIKMAKER"   <252>
157 POKE56,160:POKE44,40:POKE10240,0:NEW   <183>
   ENDE DES LISTINGS
```

```
100 REM********************************   <211>
101 REM* INTERCEPTORBASE-GRAFIKMAKER  *   <187>
102 REM*       (C)1985 BY MR-SOFT     *   <8>
103 REM*   R O N A L D    M A Y E R   *   <128>
104 REM*  A-4062  THENING 24/AUSTRIA  *   <168>
105 REM********************************   <216>
106 POKE53281,0:POKE53280,0:PRINT"{CLEAR
 GREEN}CREATING{SPACE}SPRITES{DOWN}"   <234>
107 X=64*33:S=0   <75>
108 READA$:IFA$="E"THEN112   <217>
109 A=VAL(A$):IFRIGHT$(A$,1)="X"THEN111   <135>
110 POKEX,A:X=X+1:S=S+A:GOTO108   <100>
111 FORI=0TOA-1:POKEX+I,0:NEXT:X=X+A:GOT
O108   <80>
112 IFS=42672THEN114   <42>
113 PRINT"SPRITE-DATA{SPACE}ERROR":END   <251>
114 PRINT"SPRITE-DATAS{SPACE}OK.!{DOWN}"
:GOTO139   <91>
115 REM********************************   <226>
116 REM* SPRITE-DATAS                 *   <153>
117 REM********************************   <228>
118 DATA6X,62,128,,63,170,,63,166,176,25
0,165,168,254,150,160,254,170,192,170   <208>
119 DATA168,,170,49X,10,168,7X,42,160,55
X,3,176,,1,248,,3,176,47X,10,160,,21,85   <229>
120 DATA,42,170,160,85,85,84,170,170,160
,85,85,64,170,168,,85,36X,12,48,48,55   <139>
121 DATA220,220,53,85,87,213,169,87,54,1
70,92,213,169,92,213,105,87,53,85,87,15   <12>
122 DATA215,220,,60,35X,10,128,,47,250,1
28,191,95,224,189,93,248,47,85,254,191   <217>
123 DATA85,120,191,253,248,42,255,224,,1
90,128,,40,36X,80,80,5,165,164,26,170   <124>
124 DATA169,106,187,169,26,255,169,26,25
5,228,106,190,164,102,165,164,17,144,80   <66>
125 DATA,64,35X,116,185,238,132,165,9,10
```

```
0,185,206,20,161,9,227,33,233,3X,227,37    <36>
126 DATA39,148,181,40,228,173,38,148,165
,33,227,36,206,3X,9,140,96,26,82,144,10    <203>
127 DATA82,144,10,82,144,29,140,96,14X,1
40,96,1,146,144,,146,144,,146,144,1,204    <134>
128 DATA96,49X,1,140,96,2,82,144,,146,14
4,1,18,144,3,204,96,49X,1,140,96,2,82      <217>
129 DATA144,,146,144,2,82,144,1,140,96,4
9X,3,204,96,2,18,144,1,146,144,,82,144     <93>
130 DATA3,140,96,49X,3,204,96,2,18,144,1
,146,144,,82,144,3,140,96,50X,10,160,      <32>
131 DATA170,170,,90,165,,245,95,12,15,24
0,23,1,,234,165,40,62,168,104,3,252,40     <234>
132 DATA39X,42,,174,1,106,174,5,255,234
,63,170,188,,10,160,,168,43X,192,,53,      <71>
133 DATA,13,160,,3,232,160,2,186,168,,17
4,184,,171,160,,40,128,,40,,8,37X,38,,     <188>
134 DATA170,,106,166,1,122,230,5,235,170
,255,174,128,2,170,160,,2,168,41X,255      <204>
135 DATA220,,16,,32,2,6,168,6,22,190,16
9,26,250,128,42,168,,10,224,,48,48,,15     <166>
136 DATA255,35X,223,252,,16,,32,2,6,16
8,1,22,190,166,26,250,128,42,168,,10       <53>
137 DATA224,,48,48,,15,255,48X,63,192,3,
252,47X,192,,48,,15,,3,192,,240,,          <118>
138 DATA12,47X,E                           <56>
139 REM********************************    <250>
140 REM* CHAR-DATAS                   *    <70>
141 REM********************************    <252>
142 PRINT"{DOWN}CREATING{SPACE}CHARACTER
S{DOWN}"                                   <243>
143 X=8192:S=0                             <80>
144 READA$:IFA$="E"THEN148                 <129>
145 A=VAL(A$):IFRIGHT$(A$,1)="X"THEN147    <128>
146 POKEX,A:POKEX+1024,255-A:X=X+1:S=S+A
:GOTO144                                   <199>
147 FORI=0TOA-1:POKEX+I,0:POKEX+I+1024,2
55:NEXT:X=X+A:GOTO144                      <18>
148 IFS=82881THEN150                       <101>
149 PRINT"CHARACTER-DATA{SPACE}ERROR":EN
D                                          <174>
150 PRINT"CHARACTER-DATAS{SPACE}OK.!{DOW
N}":GOTO184                                <68>
151 DATA255,255,255,255,255,4X,12,30,102
,126,204,204,,62,51,124,102,204,248,,      <67>
152 DATA30,51,96,96,204,120,,60,54,102,
102,216,240,,63,48,120,96,192,252,,63      <235>
153 DATA48,120,96,192,192,0,0,30,51,96,1
10,204,120,,51,51,126,102,204,204,,30      <12>
154 DATA12,24,24,48,120,0,0,15,6,12,12,2
16,112,,51,54,120,120,216,204,,48,48       <227>
155 DATA96,96,192,252,,49,59,127,107,19
8,198,,51,59,126,126,220,204,,30,51        <234>
156 DATA102,102,204,120,,62,51,102,124,
192,192,,30,51,102,102,120,12,,62,51       <210>
157 DATA102,124,216,204,,30,51,120,14,2
04,120,,63,12,24,24,48,48,,51,51,102       <116>
158 DATA102,204,120,,51,51,102,102,120,
48,,49,49,107,127,238,198,,51,51,60        <123>
159 DATA60,204,204,,51,51,60,24,48,48,,
,63,6,24,48,192,252,,255,4X,255,,255       <203>
160 DATA192,192,192,192,255,,255,240,24
0,240,240,255,,255,252,252,252,252,255     <182>
161 DATA,,255,255,255,255,255,255,9X,24,
24,60,60,60,255,255,255,25,43,89,100,41    <150>
162 DATA156,100,72,,1,3,7,15,63,255,255,
,,193,227,255,255,255,255,,192,224,248     <131>
163 DATA252,255,255,4X,48,126,255,255,3,
3,15,15,63,63,252,255,255,255,192,192,3    <70>
164 DATA3,15,255,240,240,240,240,192,192
,,3,3,15,15,63,63,255,255,255,240,192      <222>
165 DATA192,3,15,255,240,252,60,240,240,
192,192,,128,160,40,46,169,253,237,252     <220>
166 DATA,3,15,63,255,255,3X,64,212,245,2
53,255,3X,30,51,102,102,204,120,,12,28     <160>
167 DATA24,24,48,120,,30,51,14,56,192,2
52,,30,51,28,6,204,120,,7,15,54,126        <125>
168 DATA12,12,,63,48,124,6,204,120,,30
,48,124,102,204,120,,63,3,12,24,48,48,     <70>
169 DATA,30,51,60,102,204,120,,30,51,10
2,62,12,120,,61,61,245,245,170,190,190     <0>
170 DATA170,125,125,245,245,170,190,190,
170,112,112,224,224,160,160,128,128,,      <90>
171 DATA15,63,63,255,255,255,,240,255,25
5,255,255,255,255,,3,251,254,255,255       <121>
172 DATA255,,15,255,255,255,255,255,255,
3X,192,252,255,255,255,255,191,47,42,10    <125>
173 DATA2,,255,254,250,171,175,10,2,,25
5,255,255,255,255,255,190,170,255,255      <223>
174 DATA255,254,252,248,160,128,255,239,
191,255,190,42,,255,252,248,160,128        <56>
175 DATA128,2X,21,85,255,234,238,238,234
,255,85,85,255,238,238,238,238,255,85      <104>
176 DATA85,253,253,253,253,189,253,69,17
,17,69,1,1,1,1,16,48,16,116,84,69,68,69    <187>
177 DATA64,64,64,64,64,4X,255,85,125,85,
85,255,,255,85,85,125,85,85,85,255,255     <144>
178 DATA87,92,92,112,112,192,192,3,3,13,
13,53,53,213,255,,252,92,112,112,192       <3>
179 DATA192,,3,3,13,13,53,63,0,7,29,21,
119,117,85,117,29,240,93,215,125,85,85     <129>
180 DATA247,13,244,93,85,87,117,92,116,2
08,,255,255,255,255,,255,255,255,3X        <193>
181 DATA255,255,,255,255,3X,255,255,255
,,,255,255,255,255,,255,255,192,192,3X     <203>
182 DATA240,3,3,15,15,63,63,252,252,255,
240,192,192,3,3,15,15,63,3X,3,3,255,255    <141>
183 DATA4X,12,60,63,63,E                   <255>
184 REM********************************    <39>
185 REM* INIT-DATAS                   *    <209>
186 REM********************************    <41>
187 PRINT"{DOWN}CREATING{SPACE}HEADER{DO
WN}"                                       <97>
188 X=2048:S=0                             <114>
189 READA$:IFA$="E"THEN191                 <139>
190 A=VAL(A$):POKEX,A:X=X+1:S=S+A:GOTO18
9                                          <177>
191 IFS=1374THEN193                        <219>
192 PRINT"HEADER-DATA{SPACE}ERROR":END     <137>
193 PRINT"HEADER-DATAS{SPACE}OK.!{DOWN}"
:END                                       <189>
194 DATA,23,8,193,7,151,52,52,44,52,48,5
8,151,49,48,50,52,48,44,48,58,138,,,E      <254>
  ENDE DES LISTINGS
```

```
100 REM********************************    <211>
101 REM* INTERCEPTOR        MCODEMAKER *    <153>
102 REM*       (C)1985 BY MR-SOFT      *    <8>
103 REM*  R O N A L D    M A Y E R     *    <128>
104 REM*  A-4062  THENING 24/AUSTRIA   *    <168>
105 REM********************************    <216>
106 IFPEEK(44)<>40THENPRINT"VORHER{SPACE
}'POKE44,40'{SPACE}EINGEBEN!":END          <174>
107 FORI=6150TO8179:READA:S=S+A:POKEI,A:
NEXT                                       <43>
108 IFS=240613THENPRINT"DATAS{SPACE}OK.!
":END                                      <49>
109 PRINT"DATA{SPACE}ERROR!":END           <142>
110 DATA162,8,160,0,169,41,133,254,169,2
4,133,255,169,0,133,252,169,192            <103>
111 DATA133,253,177,254,145,252,200,208,
249,230,255,230,253,202,208,242,96         <163>
112 DATA120,169,51,162,192,141,20,3,142,
21,3,169,1,141,13,220,141,26,208           <92>
113 DATA133,2,88,96,120,169,234,162,49,1
41,21,3,142,20,3,169,129,141,13,220        <160>
114 DATA169,0,32,16,192,141,32,208,141,3
3,208,96,173,25,208,141,25,208,173         <141>
115 DATA17,208,41,127,141,17,208,166,2,1
89,132,3,141,18,208,189,137,3,141          <187>
116 DATA22,208,189,142,3,141,33,208,189,
147,3,141,27,208,224,4,208,4,169           <155>
117 DATA255,133,2,230,2,138,240,3,76,129
```

```
,234,173,60,3,240,3,32,20,193,76              <143>
118 DATA49,234,162,0,189,25,5,157,24,5,2
32,224,159,144,245,162,16,134,255             <115>
119 DATA169,0,133,254,24,172,61,3,170,17
7,254,157,63,5,138,105,40,230,255             <236>
120 DATA201,160,144,241,238,61,3,96,173,
138,3,201,208,208,13,165,2,208,252            <238>
121 DATA169,215,141,138,3,32,118,192,96,
206,138,3,96,173,140,3,201,208,208            <213>
122 DATA13,165,2,208,252,169,215,141,140
,3,32,210,192,96,206,140,3,96,173             <232>
123 DATA168,6,72,160,0,185,169,6,153,168
,6,200,192,39,208,245,104,141,207             <114>
124 DATA6,96,160,0,185,209,6,153,208,6,2
00,192,159,144,245,162,20,134,255             <83>
125 DATA169,0,133,254,24,172,62,3,170,17
7,254,157,247,6,138,105,40,230,255            <157>
126 DATA201,160,144,241,238,62,3,96,173,
141,3,201,208,208,13,165,2,240,252            <50>
127 DATA169,215,141,141,3,32,232,192,96,
206,141,3,96,173,60,3,240,23,41,1             <46>
128 DATA240,3,32,186,192,32,162,192,238,
60,3,173,60,3,208,5,169,2,141,60              <243>
129 DATA3,96,173,73,3,208,28,173,0,220,4
1,1,208,14,56,173,3,208,237,1,208             <226>
130 DATA201,108,176,3,206,1,208,173,0,22
0,41,2,208,3,238,1,208,173,0,220              <170>
131 DATA41,4,208,16,173,3,208,201,204,14
4,9,206,3,208,206,1,208,238,0,208             <88>
132 DATA173,0,220,41,8,208,16,173,3,208,
201,219,176,9,238,3,208,238,1,208             <183>
133 DATA206,0,208,173,0,220,41,16,208,28
,173,64,3,208,23,238,64,3,173,1,208           <231>
134 DATA141,5,208,173,0,208,141,4,208,17
3,21,208,9,4,141,21,208,173,0,220             <98>
135 DATA41,18,208,51,173,65,3,208,46,238
,65,3,173,1,208,141,7,208,173,3,208           <3>
136 DATA141,70,3,173,0,208,141,6,208,173
,21,208,9,8,141,21,208,162,32,142             <97>
137 DATA11,212,232,142,11,212,169,255,23
7,1,208,141,8,212,173,0,208,141,2             <22>
138 DATA208,96,173,64,3,240,53,238,4,208
,173,4,208,201,250,176,30,173,79              <171>
139 DATA3,240,38,173,78,3,41,68,201,68,2
08,29,173,12,208,56,237,4,208,201             <141>
140 DATA10,176,18,169,1,141,66,3,173,21,
208,41,251,141,21,208,169,0,141,64            <178>
141 DATA3,173,65,3,240,8,173,67,3,240,4,
206,67,3,96,169,5,141,67,3,169,255            <247>
142 DATA237,7,208,141,8,212,173,7,208,20
5,70,3,144,116,162,0,201,207,144              <177>
143 DATA8,162,40,201,215,144,2,162,80,17
3,6,208,56,233,90,74,74,74,133,255            <95>
144 DATA138,24,101,255,133,254,169,7,133
,255,160,247,140,10,212,160,0,140             <66>
145 DATA11,212,162,128,142,4,212,232,142
,4,212,202,142,4,212,177,254,201              <243>
146 DATA45,144,26,201,75,176,22,201,72,1
44,9,24,169,5,109,68,3,141,68,3,24            <184>
147 DATA169,3,109,69,3,141,69,3,185,215,
194,145,254,200,192,3,208,214,169             <19>
148 DATA0,141,65,3,173,21,208,41,247,141
,21,208,76,214,194,238,6,208,238              <220>
149 DATA7,208,96,84,85,86,173,73,3,208,3
4,162,30,189,159,7,201,28,176,22              <20>
150 DATA202,208,246,24,162,14,160,14,32,
240,255,169,42,160,195,32,30,171              <79>
151 DATA238,73,3,96,222,159,7,96,173,68,
3,240,250,173,73,3,240,8,206,73,3             <72>
152 DATA162,14,32,255,233,206,68,3,162,0
,189,160,7,201,31,144,6,232,224,30            <91>
153 DATA208,244,96,254,160,7,96,144,79,8
5,84,32,79,70,32,70,85,69,76,0,173            <53>
154 DATA78,3,41,65,201,65,208,12,173,0,2
08,105,20,237,12,208,201,40,144,53            <115>
155 DATA173,3,208,72,56,237,1,208,201,6,
144,2,104,96,104,205,1,208,240.33             <27>
156 DATA173,3,208,162,0,201,207,144,8,16
2,39,201,215,144,2,162,78,189,4,7             <76>
157 DATA201,45,144,14,201,77,176,10,169,
34,157,4,7,169,1,141,71,3,96,173              <37>
158 DATA69,3,240,28,206,69,3,162,4,189,1
22,4,201,57,240,6,254,122,4,76,135            <108>
159 DATA195,169,48,157,122,4,202,208,235
,240,223,96,173,77,3,208,76,173,1             <246>
160 DATA208,201,160,176,1,96,162,0,160,2
03,173,3,208,201,207,144,12,162,40            <126>
161 DATA160,211,201,215,144,4,162,80,160
,219,189,10,7,201,45,208,225,140              <144>
162 DATA9,208,169,166,141,8,208,173,21,2
08,9,16,141,21,208,169,244,141,6              <178>
163 DATA212,160,32,140,4,212,200,140,4,2
12,136,140,4,212,238,77,3,96,173              <96>
164 DATA78,3,41,17,201,17,208,25,173,8,2
08,237,0,208,201,20,176,15,238,71             <85>
165 DATA3,206,77,3,173,21,208,41,239,141
,21,208,96,206,9,208,174,8,208,224            <14>
166 DATA20,144,234,202,202,142,8,208,96,
173,86,3,208,30,238,86,3,169,37,141           <34>
167 DATA248,7,141,249,7,169,255,141,6,21
2,169,10,141,1,212,162,128,142,4              <166>
168 DATA212,232,142,4,212,173,248,7,201,
39,144,8,169,36,141,248,7,141,249             <54>
169 DATA7,238,248,7,238,249,7,173,1,208,
205,3,208,240,4,238,1,208,96,238              <54>
170 DATA83,3,96,173,81,3,208,1,96,173,79
,3,208,127,173,87,3,240,13,169,0              <135>
171 DATA141,87,3,32,27,198,169,46,76,170
,196,173,81,3,201,2,144,72,201,3              <15>
172 DATA144,45,240,28,173,18,208,41,1,24
0,21,32,27,198,169,50,141,254,7,173           <42>
173 DATA18,208,41,63,105,110,141,13,208,
76,229,196,173,18,208,41,2,240,8              <115>
174 DATA169,49,32,56,198,76,170,196,173,
18,208,41,1,240,16,169,211,141,13             <56>
175 DATA208,32,56,198,169,48,141,254,7,7
6,229,196,169,47,76,195,196,173,16            <0>
176 DATA208,9,64,141,16,208,169,90,141,1
2,208,173,21,208,9,64,141,21,208              <145>
177 DATA238,79,3,96,32,4,197,76,180,197,
173,16,208,41,64,208,19,173,12,208            <219>
178 DATA201,5,176,12,206,79,3,173,21,208
,41,191,141,21,208,96,173,82,3,240            <138>
179 DATA4,206,82,3,96,169,4,141,82,3,173
,12,208,208,8,173,16,208,41,191,141           <250>
180 DATA16,208,206,12,208,173,254,7,201,
50,144,36,240,5,206,254,7,176,3,238           <111>
181 DATA254,7,173,80,3,208,17,173,13,208
,205,1,208,240,10,144,4,206,13,208            <206>
182 DATA96,238,13,208,96,76,234,197,201,
49,144,26,206,12,208,173,254,7,201            <141>
183 DATA46,208,7,174,18,208,232,142,15,2
12,173,1,208,205,13,208,240,39,96             <88>
184 DATA201,48,144,229,173,12,208,41,1,2
08,3,206,13,208,173,13,208,201,120            <32>
185 DATA144,14,205,1,208,176,20,173,16,2
08,41,64,201,64,240,11,206,254,7              <39>
186 DATA173,80,3,208,3,32,234,197,96,173
,80,3,240,48,173,14,208,201,5,176             <235>
187 DATA12,173,21,208,41,127,141,21,208,
206,80,3,96,206,14,208,173,78,3,41            <227>
188 DATA129,201,129,208,17,173,14,208,56
,237,0,208,201,20,176,6,238,71,3              <135>
189 DATA76,192,197,96,173,12,208,201,90,
144,41,238,80,3,173,12,208,141,14             <44>
190 DATA208,173,13,208,141,15,208,173,21
,208,9,128,141,21,208,169,247,141             <4>
191 DATA6,212,162,128,141,4,212,232,142,
4,212,202,142,4,212,96,72,32,85,198           <121>
192 DATA162,7,189,48,198,157,13,212,202,
208,247,169,65,141,18,212,104,96              <209>
193 DATA190,0,150,100,64,0,255,72,32,85,
198,162,7,189,77,198,157,13,212,202           <145>
194 DATA208,247,169,129,141,18,212,104,9
```

```
6,0,1,0,0,128,0,255,162,7,169,0,157            <7>
195 DATA13,212,202,208,250,96,173,66,3,2
40,62,32,226,198,206,66,3,206,79               <200>
196 DATA3,169,255,141,84,3,173,12,208,14
1,10,208,173,13,208,141,11,208,173             <138>
197 DATA254,7,56,233,6,141,85,3,170,189,
180,198,141,69,3,169,37,141,253,7              <80>
198 DATA169,225,141,28,208,173,21,208,41
,191,9,32,141,21,208,173,84,3,240              <175>
199 DATA51,201,150,144,20,208,0,173,253,
7,201,39,144,5,169,36,141,253,7,238            <92>
200 DATA253,7,76,203,198,173,85,3,141,25
3,7,169,193,141,28,208,206,84,3,173            <71>
201 DATA84,3,208,8,173,21,208,41,223,141
,21,208,96,100,10,20,30,50,50,162              <185>
202 DATA7,169,0,157,255,211,202,208,250,
162,7,189,0,199,157,255,211,202,208            <61>
203 DATA247,162,129,142,4,212,202,142,4,
212,96,0,10,0,0,128,0,250,173,62               <39>
204 DATA3,201,2,176,65,201,1,240,6,169,1
,141,92,3,96,173,92,3,240,50,206               <250>
205 DATA92,3,238,88,3,173,88,3,201,2,144
,37,162,1,142,87,3,202,142,88,3,173            <180>
206 DATA81,3,201,4,144,5,169,0,141,81,3,
238,81,3,173,91,3,201,10,144,6,56              <88>
207 DATA233,2,141,91,3,96,32,8,199,173,8
9,3,208,9,32,44,193,173,76,3,141               <145>
208 DATA89,3,206,89,3,32,135,195,173,30,
208,141,78,3,173,63,3,240,6,206,63             <218>
209 DATA3,76,160,199,169,4,141,63,3,173,
74,3,240,6,206,74,3,76,146,199,173,91          <186>
210 DATA3,141,74,3,32,218,194,173,71,3,2
40,6,32,44,196,76,160,199,32,73,193,32         <44>
211 DATA255,193,32,116,196,32,55,195,32,
169,195,32,96,198,32,2,195,173,83,3            <192>
212 DATA240,1,96,173,75,3,240,6,206,75,3
,76,184,199,169,150,141,75,3,76,81,199         <25>
  ENDE DES LISTINGS


100 REM**********************************      <211>
101 REM* INTERCEPTOR    HAUPTPROGRAMM *        <105>
102 REM*       (C)1985 BY MR-SOFT      *       <8>
103 REM*    R O N A L D    M A Y E R   *       <128>
104 REM*  A-4062  THENING 24/AUSTRIA   *       <168>
105 REM**********************************      <216>
106 REM**********************************      <217>
107 REM* VARIABLES                     *       <98>
108 REM**********************************      <219>
109 SYS6150:V=53248:S=54272:JO=56320:DIM
N$(11),H$(11)                                  <116>
110 POKEV+17,0:POKEV+24,25:POKEV+32,0:PO
KEV+33,8:POKEV+34,11:POKEV+35,2                <217>
111 L$(1)="BEGINNER":L$(2)="STANDARD":L$
(3)="EXPERT{SPACE2}"                           <188>
112 DATA96,148,186,196,230                     <20>
113 DATA192,208,192,208,208                    <202>
114 DATA0,0,0,0,5                              <231>
115 DATA0,255,0,0,0                            <45>
116 FORI=0TO19:READA:POKE900+I,A:NEXT          <220>
117 SYS49175:GOSUB283:SYS49175:POKEV+17,
0                                              <57>
118 POKE911,8:POKE912,10:POKE913,4             <28>
119 FORI=1TO10:N$(I)="RONNY{SPACE12}":H$
(I)="00100":NEXT                               <217>
120 GOTO140                                    <191>
121 REM*******************************         <62>
122 REM* POINT TABLE                 *         <53>
123 REM*******************************         <64>
124 POKEV+17,0:PRINT"{CLEAR}";TAB(13)"{Y
ELLOW}TABLE{SPACE}OF{SPACE}POINTS"             <119>
125 PRINT"{DOWN3 GREEN RIGHT2 RVSON SPAC
E5 RIGHT5 SPACE10}1000{SPACE}POINTS{SPAC
E6}"                                           <189>
126 PRINT"{DOWN CYAN RIGHT2 RVSON SPACE5
 RIGHT5 SPACE11}500{SPACE}POINTS{SPACE6}
"                                              <232>
127 PRINT"{DOWN GREEN RIGHT2 RVSON SPACE
5 RIGHT5 SPACE11}300{SPACE}POINTS{SPACE6
}"                                             <171>
128 PRINT"{DOWN CYAN RIGHT2 RVSON SPACE5
 RIGHT5 SPACE11}200{SPACE}POINTS{SPACE6}
"                                              <147>
129 PRINT"{DOWN GREEN RIGHT2 RVSON SPACE
5 RIGHT5 SPACE11}100{SPACE}POINTS{SPACE6
}"                                             <115>
130 FORI=0TO4:POKEV+1+I*2,87+I*16:POKEV+
I*2,85:POKEV+39+I,14:NEXT:POKEV+21,31          <124>
131 PRINT"{DOWN CYAN RIGHT2 RVSON SPACE5
 RIGHT RVSOFF BROWN}|\ \{RVSON RIGHT CYAN
 SPACE12}90{SPACE}POINTS{SPACE6}"              <29>
132 PRINT"{DOWN GREEN RIGHT2 RVSON SPACE
5 RIGHT RVSOFF BROWN}:;<{RVSON RIGHT GRE
EN SPACE12}90{SPACE}POINTS{SPACE6}"            <59>
133 PRINT"{DOWN CYAN RIGHT2 RVSON SPACE5
 RIGHT RVSOFF BROWN}./{SPACE RVSON RIGHT
 CYAN SPACE12}60{SPACE}POINTS{SPACE6}"         <235>
134 PRINT"{DOWN GREEN RIGHT2 RVSON SPACE
5 RIGHT RVSOFF BROWN SPACE}-{SPACE RVSON
 RIGHT GREEN SPACE12}30{SPACE}POINTS{SPA
CE6}"                                          <235>
135 POKE2040,46:POKE2041,50:POKE2042,49:
POKE2043,48:POKE2044,47:POKE2045,51            <171>
136 POKEV+28,31:POKEV+37,7:POKEV+38,6:PO
KEV+22,208                                     <94>
137 POKEV+17,23:POKE198,0                      <102>
138 X=0:FORI=1TO1000:X=1-X:POKE2041,50+X
:IFPEEK(JO)>111ANDPEEK(198)=0THENNEXT          <92>
139 POKEV+21,0:RETURN                          <175>
140 REM**********************************      <81>
141 REM* MENUE                         *       <34>
142 REM**********************************      <83>
143 POKEV+17,0:SYS49175:POKEV+22,200:LE=
1                                              <89>
144 PRINT"{CLEAR YELLOW DOWN3 RIGHT11}SE
LECT{SPACE}GAME{SPACE}OPTIONS"                 <180>
145 PRINT"{DOWN6 LIG.BLUE RVSON}"TAB(11)
"{SPACE}F1{SPACE2}PLAY{SPACE}GAME{SPACES
}"                                             <229>
146 PRINT"{DOWN PURPLE RVSON}"TAB(11)"{S
PACE}F3{SPACE2}LEVEL{SPACE}BEGINNER"           <106>
147 PRINT"{DOWN GREEN RVSON}"TAB(11)"{SP
ACE}F5{SPACE2}POINT{SPACE}TABLE{SPACE3}"       <243>
148 PRINT"{DOWN ORANGE RVSON}"TAB(11)"{S
PACE}F7{SPACE2}TOP{SPACE}10{SPACE8}"           <217>
149 POKEV+17,22                                <163>
150 POKE198,0:WAIT198,1:GETA$                  <51>
151 IFA$="{F5}"THENGOSUB121:GOTO140            <226>
152 IFA$="{F7}"THENGOSUB157:GOTO140            <82>
153 IFA$="{F1}"THEN167                         <250>
154 IFA$<>"{F3}"THEN150                        <42>
155 LE=LE+1:IFLE=4THENLE=1                     <43>
156 PRINT"{HOME DOWN12 PURPLE RVSON}"TAB
(22)L$(LE):GOTO150                             <49>
157 REM**********************************      <98>
158 REM* TOPTEN                        *       <14>
159 REM**********************************      <100>
160 POKEV+17,0:SYS49175:POKEV+22,200            <188>
161 PRINT"{CLEAR YELLOW RIGHT7}INTERCEPT
OR{SPACE}BASE{SPACE3}TOPTEN{DOWN}"             <234>
162 FORI=1TO10:POKE646,I+1                     <120>
163 PRINT"{DOWN RVSON RIGHT5 SPACE}"RIGH
T$("0"+MID$(STR$(I),2),2)"{SPACE3}"N$(I)
;H$(I)"{SPACE}":NEXT                           <234>
164 POKEV+17,22:POKE198,0                      <121>
165 FORI=1TO1000:IFPEEK(JO)>111ANDPEEK(1
98)=0THENNEXT                                  <179>
166 RETURN                                     <52>
167 REM**********************************      <108>
168 REM* GAME                          *       <169>
169 REM**********************************      <110>
170 FLAG=0:PI=3                                <205>
171 SC$="00000"                                <88>
172 POKEV+17,0:SYS49175:GOSUB243:GOSUB25
2:GOSUB275                                     <10>
```



```
173 FORI=829TO860:POKEI,0:NEXT:POKE844,8
:POKE849,1:POKE828,1:POKE859,40-LE*8          <158>
174 POKEV+21,3:POKEV+17,22:SYS49152:SYS5
1025:POKEV+21,3:POKES+24,0:POKE828,0          <82>
175 SC=0:FORI=0TO4:SC=SC+(PEEK(1151-I)-4
8)*10^I:NEXTI                                 <206>
176 SC$=RIGHT$("00000"+MID$(STR$(SC),2),
5)                                            <249>
177 FORI=1TO1000:NEXT                         <82>
178 IFSC=>10000ANDFLAG=0THENFLAG=1:GOTO1
72                                            <113>
179 IFSC=>40000ANDFLAG=1THENFLAG=2:GOTO1
72                                            <173>
180 PI=PI-1:IFPI>0THEN172                     <39>
181 PRINT"{HOME DOWN14 RIGHT14 BLACK SPA
CE}GAME{SPACE}OVER{SPACE}":POKES+24,0         <72>
182 FORI=1TO1000:IFPEEK(JO)>111THENNEXT       <160>
183 POKEV+21,0:IFSC$>=H$(10)THENGOSUB192      <16>
184 GOTO140                                   <255>
185 REM********************************       <126>
186 REM* TOP-LIST CHECK                *       <115>
187 REM********************************       <128>
188 FORJ=1TO10:IFSC$<H$(J)THENNEXT            <211>
189 PRINTJ                                    <234>
190 FORK=11TOJ+1STEP-1:H$(K)=H$(K-1):N$(
K)=N$(K-1):NEXT:H$(J)=SC$:N$(J)=N$            <183>
191 RETURN                                    <77>
192 REM********************************       <133>
193 REM* NAME INPUT                    *       <214>
194 REM********************************       <135>
195 N$="":X=0:POKEV+17,0:PRINT"{CLEAR}":
SYS49175                                      <13>
196 PRINT"{DOWN5 RED}"TAB(13)"CONGRATULA
TIONS"                                        <89>
197 PRINT"{DOWN3 RIGHT3 BLUE}THIS{SPACE}
IS{SPACE}ONE{SPACE}OF{SPACE}THE{SPACE}BE
ST{SPACE}TEN{SPACE}SCORES"                    <131>
198 PRINT"{DOWN4 RIGHT3 GREEN RIGHT3}ENT
ER{SPACE}YOUR{SPACE}NAME{SPACE2 WHITE}";
:POKEV+17,22                                  <178>
199 POKE198,0:WAIT198,1:GETA$                 <100>
200 IFA$=CHR$(13)ANDX>0THEN204                <26>
201 IFA$=CHR$(20)ANDX>0THENX=X-1:PRINTCH
R$(20);:N$=LEFT$(N$,LEN(N$)-1):GOTO199        <126>
202 IFA$<"A"ORA$>"Z"ORX>14THEN199             <244>
203 PRINTA$;:N$=N$+A$:X=X+1:GOTO199           <105>
204 N$=LEFT$(N$+"{SPACE17}",17)               <78>
205 PRINT:PRINT"{DOWN2 YELLOW RIGHT6}THI
S{SPACE}IS{SPACE}TOPTEN{SPACE}POSITION{S
PACE}NO";:GOSUB185                            <143>
206 FORI=1TO1000:IFPEEK(JO)>111ANDPEEK(1
98)=0THENNEXT                                 <220>
207 RETURN                                    <93>
208 REM********************************       <149>
209 REM* SOUND                         *       <146>
210 REM********************************       <151>
211 S=54272:V=53248                           <186>
212 FORI=0TO24:POKES+I,0:NEXT                 <2>
213 POKES+03,03                               <191>
214 POKES+06,252                              <175>
215 POKES+10,4                                <58>
216 POKES+13,252                              <166>
217 POKES+17,5                                <102>
218 POKES+20,252                              <157>
219 POKES+24,15                               <233>
220 POKES+04,0:POKES+04,65                    <164>
221 POKES+11,0:POKES+11,65                    <127>
222 POKES+18,0:POKES+18,65                    <254>
223 FORJ=1TO57:READA,B,C                      <188>
224 POKES+01,A*2                              <193>
225 POKES+08,B*4                              <254>
226 POKES+15,C*8                              <31>
227 FORI=1TO70                                <158>
228 NEXTI,J                                   <164>
229 FORI=4TO18STEP7:POKES+I,64:NEXT:RETU
RN                                            <140>
230 DATA4,3,2,4,3,2,4,3,2,0,0,0,4,3,2         <227>
```



Nein, nein, so eine alte „Mühle" fliegen Sie bei INTERCEPTOR BASE nicht!

```
231 DATA4,3,2,4,4,2,0,3,0,5,4,3,5,4,3         <234>
232 DATA5,4,3,0,3,0,0,2,0,4,4,4,4,4,4,0,
3,0                                           <55>
233 DATA4,3,2,4,3,2,4,3,2,0,0,0,4,3,2         <230>
234 DATA4,3,2,4,4,2,0,3,0,5,4,3,5,4,3         <237>
235 DATA5,4,3,0,3,0,0,2,0,4,4,4,4,4,4,0,
3,0                                           <58>
236 DATA5,4,3,6,4,5,0,5,2,2,4,3               <63>
237 DATA7,4,5,0,5,2,1,4,3,4,4,3               <62>
238 DATA5,4,3,6,4,5,0,5,2,2,4,3               <65>
239 DATA7,4,6,0,5,3,1,4,4,1,4,4               <64>
240 DATA3,2,3,3,2,3,3,2,3,3,2,3               <231>
241 DATA2,3,2,2,3,2,2,3,2,2,3,2               <180>
242 DATA4,1,1                                 <39>
243 REM********************************       <184>
244 REM* SOUND                         *       <181>
245 REM********************************       <186>
246 FORI=0TO24:POKES+I,0:NEXT                 <36>
247 POKES+0,0:POKES+1,6:POKES+5,0:POKES+
6,8+16*15                                     <250>
248 POKES+7,50:POKES+8,110:POKES+12,0:PO
KES+13,0+16*10                                <162>
249 POKES+14,0:POKES+15,9:POKES+19,0          <111>
250 POKES+21,0:POKES+22,140:POKES+23,7+1
6*15                                          <161>
251 POKES+24,15+48:RETURN                     <97>
252 REM********************************       <193>
253 REM* SCREEN-GAME                   *       <255>
254 REM********************************       <195>
255 POKEV+17,0:POKE53280,0:PRINT"{CLEAR}
";                                            <175>
256 PRINT"{DOWN LIG.BLUE SPACE3}SCORE{SP
ACE5}TOP{SPACE6}PILOT{SPACE4}LEVEL{SPACE
4}";                                          <99>
257 PRINT"{DOWN ORANGE SPACE3}"SC$"{SPAC
E4}"H$(1)"{SPACE6}"PI"{SPACE4}"L$(LE)"{S
PACE2}";                                      <224>
258 PRINT"{BLACK DOWN}@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@";                     <104>
259 PRINT"{HGREY DOWN SPACE5}=>?->♠{SPAC
E9}=>♠{SPACE17}";                             <249>
260 PRINT"{SPACE5}|-‾_|=>->♠{SPACE4}|‾_
{SPACE6}=>???->♠{SPACE3}";                    <229>
261 PRINT"{SPACE11}|-‾|{SPACE13}|-‾-‾_
|{SPACE3}";                                   <235>
262 PRINT"{SPACE40}";                         <62>
263 PRINT"{DOWN ORANGE}00000000000000000
00000000000000000000000";                     <188>
264 PRINT"♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣♣
♣♣♣♣♣♣♣♣♣♣";                                  <32>
```

```
265 PRINT"{DOWN PURPLE}||||||||||||||||
|||||||||||||||||||||||||";                 <210>
266 PRINT"♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦
♦♦♦♦♦♦♦♦♦♦♦";                               <42>
267 PRINT"{BLUE}!#$%&#%&&!#$%&!!#%&#$%!#
%!#$%&#%&&!!!#$%";                          <117>
268 PRINT"{BROWN SPACE40}";                 <96>
269 PRINT"{SPACE2}●┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┐'()'(
)'+)'+)●┌┬┬┬┬┬┬┐";                          <174>
270 PRINT"{SPACE}●┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬*+,▒|,
◥π,*+{SPACE}●┌┬┬┬┬┬┬┐{SPACE}";              <111>
271 PRINT"{SPACE40}";                       <71>
272 PRINT"{BLACK}@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@";                         <213>
273 PRINT"{CYAN SPACE2}FUEL{SPACE2 RED}_
_________{PURPLE}__________{LIG.RED}____
______";                                    <116>
274 RETURN                                  <161>
275 REM*******************************      <217>
276 REM* SPRITES                     *      <132>
277 REM*******************************      <219>
278 POKEV,99:POKEV+1,210:POKEV+2,99:POKE
V+3,211:POKEV+37,1:POKEV+38,0               <206>
279 POKEV+28,65:POKEV+39,2:POKEV+40,0:PO
KEV+41,0:POKEV+42,11:POKEV+43,0             <7>
280 POKEV+44,0:POKEV+45,3:POKEV+46,0        <188>
281 POKE2040,33:POKE2041,36:POKE2042,34:
POKE2043,35:POKE2044,53:POKE2047,52         <83>
282 RETURN                                  <169>
283 REM*******************************      <225>
284 REM* TITLE                       *      <191>
285 REM*******************************      <227>
286 PRINT"{CLEAR DOWN2}"                    <26>
287 PRINTTAB(15)"{YELLOW}INTERCEPTOR{DOW
N}"                                         <121>
288 PRINTTAB(14)"{BLUE SPACE}'()'()'+)'+
)"                                          <80>
289 PRINTTAB(14)"{RED}*+,▒|,◥π,*+"          <145>
290 PRINT"{RIGHTS DOWN3 GREEN}COPYRIGHT{
SPACE}1985{SPACE}BY{SPACE}RONALD{SPACE}M
AYER"                                       <129>
291 PRINT"{CYAN DOWN3 SPACE11}PRESS{SPAC
E}FIRE{SPACE}TO{SPACE}START{DOWN2}":POKE
903,228:POKEV+17,22                         <25>
292 FORI=828TO860:POKEI,0:NEXT              <59>
293 SYS49152:FORI=228TO196STEP-.1:POKE90
3,I:NEXT                                    <167>
294 PRINT"{BLACK RVSON SPACE40 RVSOFF HG
REY}";                                      <78>
295 PRINT"{SPACE40}";                       <95>
296 PRINT"{SPACE4}./././././././././././
./././././././";                            <94>
297 PRINT"{SPACE3}●┌┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬
┬┬┬┬┬┬┬┬┬┬┬┬┐{SPACE}";                      <232>
298 PRINT"{SPACE2}./././././././././././
./././././././{SPACE2}";                    <76>
299 POKE828,1:POKEV+21,3:GOSUB275:POKEV+
1,200                                       <77>
300 GOSUB208                                <142>
301 FORI=1TO3000:IFPEEK(JO)>111THENNEXT     <36>
302 POKEV+21,0:RETURN                       <83>
  ENDE DES LISTINGS
```

INTERCEPTOR-BASE — THE END

Ein Geschicklichkeitsspiel, das durch seine hervorragende Grafik besticht

Das Abtippen: Die Programme werden in der Reihenfolge, wie sie abgedruckt sind, eingetippt und abgespeichert. Anschließend werden die ersten drei eingeladen und gestartet. Nun wird das vierte Programm (Hauptprogramm) geladen. Dann gibt man "POKE 45, 0: POKE 46,96:CLR ein und speichert es wieder ab. Dieses Programm ist die fertige Spielversion. Wenn man spielen will, lädt man also nur noch dieses Programm ein. Die anderen drei Programme können jetzt gelöscht werden.



„Tach, Herr Tronic! Ich komme, um den Schotter abzuholen (greifer, gier)!"

Vorgeschichte: Sie haben nach unsäglichen Mühen und Entbehrungen ein Programm zustandegebracht. Dieses Programm schicken Sie an den Tronic-Verlag, um es veröffentlichen zu lassen und vor allem, um das Honorar einzustreichen.
Nach Wochen gespannten Wartens kommt schließlich die tolle Nachricht: Ihr Programm wird veröffentlicht. Ihr Honorar aber, und damit kommen wir zum Spiel, müssen Sie sich selbst „zusammenklauben". D.h.: Sie fahren vor der Wehretaler Skyline mit Ihrem Säckel hin und her und sammeln die Knete ein.
So weit, so einfach. Damit Sie aber den Tronic-Verlag nicht bis aufs letzte Hemd ausziehen, ist eine Schwierigkeit eingebaut: Fällt zuviel Schotter (Aha, da is' es) neben Ihr hoffentlich prall gefülltes Säkkel, ist Schluß. Dann wird der Tresor dichtgemacht.
Also ran an den Joystick, ran ans Geld!

Schotter – Teil Eins

```
C64 LISTING MIT CHECKSUMMEN (C V1.0)

106 REM ZEICHEN/SPRITES ZU SCHOTTER             <47>
107 :                                           <165>
202 :                                           <4>
203 DATA1,0,24,60,102,102,126,102,0             <76>
204 DATA2,0,124,102,124,102,102,124,0           <109>
205 DATA3,0,60,102,96,96,102,60,0               <15>
206 DATA4,0,120,108,102,102,108,120,0           <125>
207 DATA5,0,126,96,124,96,96,126,0              <165>
208 DATA6,0,126,96,124,96,96,96,0               <30>
209 DATA7,0,62,96,96,110,102,62,0               <233>
210 DATA8,0,102,102,102,126,102,102,0           <43>
211 DATA9,0,126,24,24,24,24,126,0               <220>
212 DATA10,0,6,6,6,6,102,60,0                   <184>
213 DATA11,0,102,108,120,120,108,102,0         <79>
214 DATA12,0,96,96,96,96,96,126,0              <10>
215 DATA13,0,99,119,127,107,99,99,0             <19>
216 DATA14,0,102,118,126,126,110,102,0         <155>
217 DATA15,0,60,102,102,102,102,60,0           <119>
218 DATA16,0,124,102,102,124,96,96,0           <224>
219 DATA17,0,60,102,102,102,110,62,0           <156>
220 DATA18,0,124,102,102,124,108,102,0         <149>
221 DATA19,0,62,96,60,6,6,124,0                 <111>
222 DATA20,0,126,24,24,24,24,24,0              <152>
223 DATA21,0,102,102,102,102,102,60,0          <252>
224 DATA22,0,102,102,102,102,60,24,0           <167>
225 DATA23,0,99,99,107,127,119,99,0            <194>
226 DATA24,0,102,102,60,60,102,102,0           <127>
227 DATA25,0,102,102,60,24,24,24,0             <103>
228 DATA26,0,126,12,24,48,96,126,0             <8>
229 DATA32,0,0,0,0,0,0,0                        <132>
230 DATA33,0,24,24,24,24,0,24,0                <130>
231 DATA39,24,16,0,0,0,0,0                      <119>
232 DATA40,0,24,48,48,48,48,24,0               <13>
233 DATA41,0,24,12,12,12,12,24,0               <239>
234 DATA44,0,0,0,0,0,24,16                      <57>
235 DATA46,0,0,0,0,0,24,0                       <190>
236 DATA48,0,60,102,102,102,102,60,0           <153>
237 DATA49,0,24,56,24,24,24,126,0              <29>
238 DATA50,0,60,102,12,24,48,126,0             <22>
239 DATA51,0,126,12,24,12,102,60,0             <40>
240 DATA52,0,12,28,60,108,126,12,0             <232>
241 DATA53,0,126,96,124,6,102,60,0             <255>
242 DATA54,0,60,96,124,102,102,60,0            <222>
243 DATA55,0,126,6,12,24,48,48,0               <175>
244 DATA56,0,60,102,60,102,102,60,0            <87>
245 DATA57,0,60,102,62,6,12,56,0               <242>
246 DATA128,0,0,0,0,0,0,127                     <57>
247 DATA129,255,195,3,3,3,3,3,3                 <36>
248 DATA130,3,3,3,3,3,3,3,7                     <190>
249 DATA131,0,0,0,0,0,0,248                     <119>
250 DATA132,252,12,0,0,56,56,24,24,133,2
4,24,27,31,30,28,24,185                         <92>
251 DATA134,0,0,143,223,216,216,223,207         <145>
252 DATA135,0,0,224,225,225,0,225,192           <137>
```

```
253 DATA136,0,0,254,254,128,6,254,252          <83>
254 DATA137,0,0,127,255,199,192,255,126        <148>
255 DATA138,0,0,63,127,96,96,127,63            <223>
256 DATA139,0,0,159,191,48,48,191,159          <211>
257 DATA140,0,0,155,223,220,216,216,136        <161>
258 DATA141,0,0,231,239,108,108,111,103        <225>
259 DATA142,48,48,240,240,48,48,240,240        <53>
260 DATA143,0,0,0,0,112,112,48,48              <100>
261 DATA144,0,0,127,255,195,195,255,127        <171>
262 DATA145,96,96,96,224,192,0,0,0             <118>
263 DATA146,0,0,108,125,113,97,97,32           <248>
264 DATA147,0,0,252,254,134,134,254,252        <48>
265 DATA148,0,0,71,207,204,204,239,103         <92>
266 DATA149,0,0,0,0,64,192,64,0                <111>
267 DATA150,96,96,96,224,192,0,0,0             <106>
268 DATA151,6,6,246,247,119,7,246,242          <236>
269 DATA152,0,0,0,0,14,14,6,6                  <245>
270 DATA153,63,12,236,236,140,140,205,10
2                                              <161>
271 DATA154,0,0,0,0,12,12,12,28                <185>
272 DATA155,24,24,24,24,60,60,36,36            <134>
273 DATA156,36,102,66,66,195,129,129,24        <159>
274 DATA58,0,0,0,16,0,16,0,0                   <41>
275 DATA59,15,63,127,127,255,255,255,255       <46>
276 DATA60,255,255,255,255,127,127,63,15       <109>
277 DATA61,255,255,255,255,254,254,252,2
40                                             <144>
278 DATA62,240,252,254,254,255,255,255,2
55                                             <136>
279 DATA63,240,192,128,128,0,0,0,0             <89>
280 DATA64,0,0,0,0,128,128,192,240            <229>
281 DATA65,15,3,1,1,0,0,0,0                    <189>
282 DATA66,0,0,0,0,1,1,3,15                    <82>
283 DATA96,0,0,0,0,0,0,0,0                     <211>
284 DATA43,255,255,255,255,255,255,255,2
55                                             <249>
285 DATA47,0,0,16,34,182,246,255,255           <34>
286 DATA108,0,255,0,60,0,0,255,0               <138>
287 DATA85,255,255,0,0,0,0,0,0                 <80>
294 DATA73,1,1,1,1,1,1,1,1                     <49>
296 DATA74,136,136,136,255,136,136,136,2
55                                             <66>
297 DATA75,192,192,192,192,192,192,192,1
92                                             <2>
298 DATA76,0,0,0,31,4,4,4,255                  <138>
299 DATA77,0,0,0,125,69,125,73,255             <230>
300 DATA78,0,0,0,244,22,21,244,255            <242>
301 DATA79,0,0,0,190,136,136,190,255          <181>
302 DATA80,0,0,0,248,128,128,248,255          <99>
303 DATA81,0,0,0,0,0,0,0,255                  <121>
999 DATA-1                                     <90>
1000 REM ********** SPRITES ************       <194>
1001 DATA0,16,0,0,40,0,0,40,0,0,68,0,1,2
55,0,0,56,0,0,56,0,0,56,0,0,56,0,0,56          <248>
1002 DATA0,0,56,0,0,56,0,0,56,0,0,56,0,0
,56,0,0,56,0,0,124,0,0,124,0                   <59>
1003 DATA0,124,0,0,124,0,0,124,0               <251>
1004 :                                         <41>
1005 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,1,8,0,1,240,0,0,224,0             <207>
1006 DATA0,248,0,0,240,0,3,184,0,7,28,0,
15,62,0,15,30,0,15,158,0                       <200>
1007 DATA15,30,0,7,188,0,3,248,0,0,0           <157>
1008 :                                         <45>
1009 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1                 <49>
1010 DATA224,0,7,248,0,15,252,0,30,222,0
,28,158,0,62,223,0,62,223,0,30                 <125>
1011 DATA222,0,30,222,0,15,252,0,7,248,0
,1,224                                         <157>
1012 :                                         <49>
5001 READD:IFD=-1THEN5003                      <18>
5002 FORTT=0TO7:READDD:POKE14336+D*8+TT,
DD:NEXT:GOTO5001                               <87>
5003 FORT=255TO253STEP-1                       <151>
5004 FORTT=0TO62:READD:POKET*64+TT,D:NEX
T:NEXT                                         <245>
```

Schotter – Zweiter Teil

```
0 DATA169,0,160,0,141,232,3,141,233,3,14
1,234,3,141,236,3,153,0,212,200                <121>
1 DATA192,24,208,248,169,31,141,24,212,1
69,241,141,23,212,169,120,141,6                <30>
2 DATA212,169,240,141,13,212,141,20,212,
169,33,141,4,212,169,100,141,235               <75>
3 DATA3,169,17,141,11,212,169,101,141,18
,212,76,41,81,88,96,32,180,80,172              <82>
4 DATA232,3,192,12,240,7,200,140,232,3,7
6,49,234,160,0,140,232,3,174,233               <141>
5 DATA3,224,64,208,13,162,0,142,233,3,17
3,236,3,73,255,141,236,3,189,217               <100>
6 DATA80,141,1,212,232,189,217,80,141,0,
212,232,142,233,3,173,4,220,77                 <214>
7 DATA5,220,41,7,10,168,190,25,81,185,26
,81,32,165,80,142,8,212,141,7,212              <193>
8 DATA142,15,212,141,14,212,76,49,234,44
,236,3,16,6,10,168,138,42,170,152              <231>
9 DATA96,76,49,234,172,234,3,192,4,240,5
,200,140,234,3,96,160,0,140,234                <52>
10 DATA3,172,235,3,200,208,2,160,100,140
,22,212,140,235,3,169,190,141,7                <8>
11 DATA212,96,4,226,9,196,4,226,9,196,4,
226,9,196,4,226,9,196,4,90,8,182               <236>
12 DATA4,90,8,180,4,90,8,180,4,90,8,180,
6,133,13,10,6,133,13,10,6,133,13               <23>
13 DATA10,6,133,13,10,5,207,11,158,5,207
,11,158,5,123,10,247,5,123,10,247              <146>
14 DATA19,137,21,237,23,59,26,20,29,69,1
7,103,19,137,21,237,173,18,208,201             <204>
15 DATA242,208,249,169,7,141,22,208,169,
0,133,251,133,252,169,0,162,64,133             <17>
16 DATA253,134,254,141,0,192,169,1,162,0
,157,192,219,232,224,39,208,248                <60>
17 DATA120,169,113,160,81,141,20,3,140,2
1,3,169,242,141,18,208,173,17,208              <37>
18 DATA41,127,141,17,208,169,129,141,26,
208,88,96,173,25,208,141,25,208                <232>
19 DATA48,7,173,13,220,88,76,72,80,173,0
,220,201,111,240,18,173,18,208,201             <8>
20 DATA250,176,36,32,247,81,169,250,141,
18,208,76,129,234,120,173,17,208               <76>
21 DATA41,127,141,17,208,169,240,141,26,
208,169,49,162,234,141,20,3,142                <173>
22 DATA21,3,88,165,252,240,52,162,0,189,
193,7,157,192,7,232,224,38,208,245             <195>
23 DATA172,0,192,177,253,201,127,240,14,
141,230,7,200,140,0,192,208,21,230             <20>
24 DATA254,76,234,81,169,0,162,64,133,25
3,141,0,192,134,254,169,32,141,230             <55>
25 DATA7,169,241,141,18,208,169,200,141,
22,208,76,129,234,169,0,133,252                <38>
26 DATA165,251,201,2,208,2,230,252,170,2
02,202,138,41,7,133,251,141,22,208             <239>
27 DATA96,169,0,32,41,82,169,11,32,41,82
,169,12,32,41,82,169,15,32,41,82               <42>
28 DATA169,1,32,41,82,96,160,0,153,208,2
18,200,192,39,208,248,160,0,162                <84>
29 DATA0,232,208,253,200,192,40,208,246,
96,253,169,0,141,32,208,141,33,208             <226>
30 DATA169,11,141,17,208,76,144,85,147,5
,83,67,72,79,84,84,69,82,33,46,32              <150>
31 DATA71,69,70,65,76,76,69,78,32,58,32,
32,32,32,80,85,78,75,84,69,32,58               <255>
32 DATA32,32,32,32,32,32,32,32,32,32,32,
32,32,46,32,32,32,32,32,32,32,32               <61>
33 DATA32,32,32,32,32,46,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32                  <198>
34 DATA32,32,32,32,32,32,32,32,32,32,46,
32,32,32,32,32,32,32,32,32,32,32               <18>
35 DATA32,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,46,32,32,32,32,32,32               <94>
36 DATA32,32,46,32,32,32,32,32,32,32,32,
```

```
32,32,32,46,32,32,32,32,32,32,32                 <124>
37 DATA32,32,32,46,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,32                 <172>
38 DATA32,32,32,32,32,32,32,46,32,32,32,
32,32,32,32,32,32,32,32,32,32,32                 <233>
39 DATA32,32,46,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,46,32,32,32                 <187>
40 DATA32,32,32,32,32,32,32,32,32,32,32,
46,32,32,32,32,32,32,32,32,32,32                 <39>
41 DATA32,32,32,32,32,46,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,32                 <206>
42 DATA46,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,46,32                 <190>
43 DATA32,32,32,32,32,32,32,32,32,32,46,
32,32,32,32,32,32,32,32,32,32,32                 <27>
44 DATA32,32,32,32,32,32,32,32,32,46,32,
32,32,32,32,32,32,32,32,32,46,32                 <71>
45 DATA32,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,46,32,32,32,32,32,32                 <104>
46 DATA32,32,46,32,32,32,32,32,32,32,32,
46,32,32,32,32,32,32,32,32,32,32                 <89>
47 DATA32,32,32,32,46,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,46,32,32                 <240>
48 DATA32,32,32,32,32,32,32,32,32,32,32,
32,46,32,32,32,46,32,32,32,32,32                 <60>
49 DATA32,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,32                 <125>
50 DATA46,32,32,32,32,32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,32,32                 <140>
51 DATA32,32,32,32,46,32,32,32,32,32,32,
32,32,32,46,32,32,32,32,32,32,32                 <169>
52 DATA32,32,46,32,32,32,32,32,32,32,32,
156,204,208,209,209,209,209,32,32                <34>
53 DATA159,209,32,32,32,32,32,158,204,20
5,206,207,208,32,32,5,205,208,209                <190>
54 DATA209,209,32,32,32,32,32,32,32,32,3
2,32,32,32,156,32,201,202,202,202                <61>
55 DATA202,202,202,203,159,201,202,203,5
,32,209,32,152,201,202,202,202,202               <0>
56 DATA202,203,5,201,202,202,202,202,202
,203,158,32,205,208,209,209,209                  <10>
57 DATA209,32,32,32,32,32,156,201,202,20
2,202,202,202,202,203,159,201,202                <166>
58 DATA203,5,201,202,203,152,201,202,202
,202,202,202,203,5,201,202,202,202               <116>
59 DATA202,202,203,158,201,202,202,202,2
02,202,202,203,32,32,32,32,156,201               <82>
60 DATA202,202,202,202,202,202,203,159,2
01,202,203,5,201,202,203,152,201                 <17>
61 DATA202,202,202,202,202,203,5,201,202
,202,202,202,202,203,158,201,202                 <245>
62 DATA202,202,202,202,202,203,151,32,17
2,172,172,172,172,172,172,172,172                <87>
63 DATA172,172,172,172,172,172,172,172,1
72,172,172,172,172,172,172,172,172               <1>
64 DATA172,172,172,172,172,172,172,172,1
72,172,172,172,172,172,30,47,47                  <20>
65 DATA47,47,47,47,47,47,47,47,47,47,47,
47,47,47,47,47,47,47,47,47,47,47                 <65>
66 DATA47,47,47,47,47,47,47,47,47,47,47,
47,47,47,47,47,156,213,213,213                   <146>
67 DATA213,213,213,213,213,213,213,213,2
13,213,213,213,213,213,213,213,213               <185>
68 DATA213,213,213,213,213,213,213,213,2
13,213,213,213,213,213,213,213,213               <186>
69 DATA213,213,5,17,17,32,32,32,32,32,32
,32,32,32,32,32,84,72,69,32,83,69                <56>
70 DATA67,79,78,68,32,80,82,79,74,69,75,
84,32,32,32,32,32,32,32,32,32,32                 <222>
71 DATA32,191,169,81,162,82,133,251,134,
252,160,0,177,251,201,191,240,11                 <68>
72 DATA32,210,255,200,208,244,230,252,76
,154,85,169,27,141,17,208,32,180                 <47>
73 DATA85,96,173,18,208,201,240,208,249,
169,7,141,22,208,169,0,133,251,133               <57>
74 DATA252,141,0,192,133,182,141,3,192,1
41,8,192,169,4,133,252,120,169,244               <140>
75 DATA160,85,141,20,3,140,21,3,169,146,
141,18,208,173,17,208,41,127,141                 <167>
76 DATA17,208,169,129,141,26,208,88,96,1
73,25,208,141,25,208,48,7,173,13                 <125>
77 DATA220,88,76,49,234,173,8,192,201,1,
240,86,173,18,208,201,186,176,104                <8>
78 DATA173,0,220,201,119,240,50,201,118,
240,46,201,117,240,42,173,0,220                  <24>
79 DATA201,123,240,46,201,122,240,42,201
,121,240,38,173,235,3,201,255,240                <44>
80 DATA20,201,1,240,27,32,48,88,169,0,14
1,27,208,169,186,141,18,208,76,129               <162>
81 DATA234,32,3,88,169,0,141,234,3,76,61
,86,32,82,88,169,1,141,234,3,76                  <109>
82 DATA61,86,120,173,17,208,41,127,141,1
7,208,169,240,141,26,208,169,49                  <60>
83 DATA160,234,141,20,3,140,21,3,88,173,
18,208,201,199,176,66,173,0,220                  <36>
84 DATA201,119,240,39,201,118,240,35,201
,117,240,31,173,0,220,201,123,240                <103>
85 DATA30,201,122,240,26,201,121,240,22,
32,59,88,169,2,141,27,208,169,199                <227>
86 DATA141,18,208,76,129,234,32,242,87,7
6,161,86,32,65,88,76,161,86,169                  <140>
87 DATA199,141,18,208,76,129,234,169,146
,141,18,208,169,200,141,22,208,165               <23>
88 DATA189,208,6,76,129,234,76,108,87,17
3,234,3,201,0,208,246,201,1,208                  <253>
89 DATA3,76,129,234,162,0,189,225,5,157,
224,5,189,225,217,157,224,217,232                <71>
90 DATA224,39,208,239,162,0,189,9,6,157,
8,6,189,9,218,157,8,218,232,224                  <208>
91 DATA39,208,239,162,0,189,49,6,157,48,
6,189,49,218,157,48,218,232,224                  <18>
92 DATA39,208,239,162,0,189,89,6,157,88,
6,189,89,218,157,88,218,232,224                  <99>
93 DATA39,208,239,162,0,173,224,5,141,7,
6,173,8,6,141,47,6,173,48,6,141                  <126>
94 DATA87,6,173,88,6,141,127,6,173,224,2
17,141,7,218,173,8,218,141,47,218                <8>
95 DATA173,48,218,141,87,218,173,88,218,
141,127,218,169,0,141,233,3,133                  <49>
96 DATA189,76,129,234,162,38,189,224,5,1
57,225,5,189,224,217,157,225,217                 <69>
97 DATA202,224,255,208,239,162,38,189,8,
6,157,9,6,189,8,218,157,9,218,202                <192>
98 DATA224,255,208,239,162,38,189,48,6,1
57,49,6,189,48,218,157,49,218,202                <54>
99 DATA224,255,208,239,162,38,189,88,6,1
57,89,6,189,88,218,157,89,218,202                <231>
100 DATA224,255,208,239,173,7,6,141,224,
5,173,47,6,141,8,6,173,87,6,141,48               <92>
101 DATA6,173,127,6,141,88,6,173,7,218,1
41,224,217,173,47,218,141,8,218,173              <138>
102 DATA87,218,141,48,218,173,127,218,14
1,88,218,169,0,141,233,3,133,189                 <111>
103 DATA76,129,234,165,251,170,202,202,1
38,41,7,133,251,141,22,208,32,246                <104>
104 DATA88,96,169,0,133,189,173,235,3,20
1,255,208,9,165,252,240,5,169,0,141              <155>
105 DATA235,3,165,252,201,1,208,5,230,18
9,206,235,3,170,202,138,41,7,133                 <236>
106 DATA252,141,22,208,32,127,88,96,165,
252,141,22,208,169,10,141,234,3,96               <215>
107 DATA165,251,141,22,208,96,165,251,17
0,232,232,138,41,7,133,251,141,22                <106>
108 DATA208,32,42,89,96,169,0,133,189,17
3,235,3,240,11,165,252,201,7,240                 <165>
109 DATA5,169,0,141,235,3,165,252,201,6,
208,5,230,189,238,235,3,170,232,138              <48>
110 DATA41,7,133,252,141,22,208,32,142,8
8,96,172,6,192,32,157,88,136,136                 <136>
111 DATA140,0,208,140,6,192,96,172,6,192
,32,202,88,200,200,140,0,208,140                 <140>
112 DATA6,192,96,173,16,208,41,1,201,1,2
```

```
40,19,192,1,240,1,96,238,16,208,24        <240>
113 DATA169,91,141,0,208,141,6,192,168,9
6,192,1,240,1,96,206,16,208,169,255       <55>
114 DATA141,0,208,141,6,192,96,173,16,20
8,41,1,201,1,240,17,192,255,240,1         <36>
115 DATA96,238,16,208,169,1,141,0,208,14
1,6,192,96,192,91,240,1,96,206,16         <233>
116 DATA208,169,1,141,0,208,141,6,192,16
8,96,174,4,208,173,16,208,41,4,201        <26>
117 DATA4,240,10,224,2,144,77,202,202,14
2,4,208,96,224,2,176,246,173,16,208       <184>
118 DATA201,5,240,8,169,0,141,16,208,76,
36,89,169,1,141,16,208,169,255,141        <231>
119 DATA4,208,96,174,4,208,173,16,208,41
,4,201,4,208,10,224,81,176,25,232         <238>
120 DATA232,142,4,208,96,224,254,144,246
,173,16,208,24,105,4,141,16,208,169       <3>
121 DATA1,141,4,208,96,169,1,141,232,3,9
6,234,234,234,234,234,234,234,234,-1      <248>
122 MS=20480                              <123>
123 READD:IFD=-1THEN200                   <60>
124 POKEMS,D:MS=MS+1:A=A+D:GOTO123        <176>
200 IFA<>263207THENPRINT"{CLEAR}DATA{SPA
CE}ERROR{SPACE}!!"                        <10>
  ENDE DES LISTINGS
```

Schotter – Dritter Teil

```
0 :                                       <58>
1 :                                       <59>
3 REM      TEXT ERZEUGER ZU               <210>
4 REM           SCHOTTER                  <251>
5 :                                       <63>
6 :                                       <64>
10 PRINT"{CLEAR}(C) 1985{SPACE}THE{SPACE
}SECOND{SPACE}PROJEKT{SPACE}..........{S
PACE2}"                                   <107>
11 PRINT"{UP SPACE40}"                    <243>
20 PRINT"{UP}HIER{SPACE}IST{SPACE}ALSO{S
PACE}'SCHOTTER'.{SPACE16}"                <203>
21 PRINT"{UP SPACE40}"                    <253>
25 PRINT"{UP}KEINE{SPACE}ANGST,{SPACE}DA
S{SPACE}HAT{SPACE}NICHTS{SPACE}MIT{SPACE
}EISENBAH"                                <204>
30 PRINT"{UP}NEN{SPACE}ZU{SPACE}TUN{SPAC
E}!{SPACE}BEI{SPACE}SCHOTTER{SPACE}GEHT{
SPACE}ES{SPACE}UM{SPACE}FOL"              <13>
40 PRINT"{UP}GENDES:{SPACE}SIE{SPACE}HAB
EN{SPACE}EIN{SPACE}PROGRAMM{SPACE}GESCHR
IEB"                                      <220>
45 PRINT"{UP}EN{SPACE}UND{SPACE}ES{SPACE
}DEM{SPACE}TRONIC{SPACE}VERLAG{SPACE}GES
CHICKT.{SPACE}D"                          <118>
50 PRINT"{UP}ER{SPACE}VERLAG{SPACE}ERKLA
ERT{SPACE}SICH{SPACE}BEREIT,{SPACE}DIESE
S{SPACE}P"                                <167>
55 PRINT"{UP}ROGRAMM{SPACE}ZU{SPACE}VERO
EFFENTLICHEN.{SPACE}IHR{SPACE}HONORAR"    <44>
60 PRINT"{UP SPACE}ABER,{SPACE}DAS{SPACE
}IST{SPACE}DAS{SPACE}GEMEINE,{SPACE}MUES
SEN{SPACE}SIE{SPACE}"                     <29>
65 PRINT"{UP}SICH{SPACE}SELBST{SPACE}BES
ORGEN.{SPACE}D.H.{SPACE}SIE{SPACE}MUESSE
N{SPACE}D"                                <222>
70 PRINT"{UP}AS{SPACE}GELD{SPACE}AUFFANG
EN{SPACE}OHNE{SPACE}DAS{SPACE}ETWAS{SPAC
E}RUNTERF"                                <132>
75 PRINT"{UP}AELLT.{SPACE}DENN{SPACE}FAE
LLT{SPACE}MEHR{SPACE}ALS{SPACE}ERLAUBT{S
PACE}RUNT"                                <56>
80 PRINT"{UP}ER{SPACE}IS{SPACE}SCHLUSS{S
PACE}MIT{SPACE}KNETE{SPACE}!!{SPACE}ALSO
{SPACE}VORSICHT"                          <254>
85 PRINT"{UP SPACE}UND{SPACE}NUN{SPACE}V
OR{SPACE}ALLEM{SPACE}'VIEL{SPACE}SPASS'{
SPACE}........{SPACE2}"                   <107>
90 PRINT"{UP SPACE40}"                    <66>
95 PRINT"{UP}ACH{SPACE}SO{SPACE}JA,{SPAC
E}DIE{SPACE}STEUERUNG:{SPACE}JOYSTICK{SP
ACE}PORT{SPACE}"                          <41>
96 PRINT"{UP}II{SPACE}...............{
SPACE20}"                                 <171>
97 PRINT"{UP SPACE40}"                    <73>
98 PRINT"{UP}"                            <152>
99 FORT=0TO21*40:POKE4*4096+T,PEEK(1024+
T):NEXT                                   <130>
100 PRINT"{CLEAR}"                        <130>
  ENDE DES LISTINGS
```

Schotter – Vierter Teil

```
0 HI=100                                  <196>
1 :                                       <59>
2 REM (C) 1985 THE SECOND PROJEKT         <23>
3 REM                                     <146>
4 REM       -- SCHOTTER --                <25>
5 :                                       <63>
10 POKE53280,9:POKE53281,9:SI=54272:V=53
248:IFPO>HITHENHI=PO                      <125>
15 PO=0:AG=0:BE=0:G=0:GF=0:FA=0:POKE5327
2,31:POKE53265,11                         <167>
20 PRINT"{CLEAR YELLOW SPACE19 RVSON}["   <105>
25 PRINT"{SPACE11 RVSON}@C{RVSOFF SPACE6
 RVSON}\"                                 <20>
30 PRINT"{SPACE11 RVSON}AD{RVSOFF SPACE8
 RVSON}O{RVSOFF SPACE3 RVSON}UXZ"         <148>
35 PRINT"{SPACE11 RVSON}BEFGHIJKLMNPRSTW
Y                                         <95>
40 PRINT"{SPACE22 RVSON}Q{RVSOFF SPACE2
RVSON}V"                                  <109>
45 PRINT"{SPACE15}PRESENTS:               <130>
50 PRINT"{BLACK DOWN SPACE}*>{SPACE};+>{
SPACE};+>{SPACE};?>{SPACE};+>*++>*+++?;+
>{SPACE};+>*>                             <88>
55 PRINT"{SPACE2}?{SPACE}+?*{SPACE}+?*{S
PACE}+|+{SPACE}+?+{SPACE}*+?{SPACE}*+?{S
PACE}+?*{SPACE}+-+{SPACE}?                <250>
60 PRINT"{SPACE4}<+>{SPACE}+{SPACE3}+++{
SPACE}+{SPACE}+{SPACE2}+{SPACE3}+{SPACE2
}+=?{SPACE}++=                            <114>
65 PRINT"{SPACE4}-|+{SPACE}+-|{SPACE}+?+
{SPACE}+|+{SPACE2}+{SPACE3}+{SPACE2}+-|{
SPACE}+<-                                 <88>
70 PRINT"{SPACE4}<+={SPACE}<+={SPACE}<|=
{SPACE}<+={SPACE}|+-{SPACE}|+-{SPACE}<+=
{SPACE}<*>                                <230>
745 PRINT"{WHITE DOWN3 SPACE4}GESCHRIEBE
N{SPACE}VON{SPACE}:{SPACE}EDZARD{SPACE}W
ITTIG                                     <246>
750 PRINT"{BLACK DOWN2 SPACE8}LETZTER{SP
ACE}HIGHSCORE{SPACE}:";HI                 <120>
755 POKE53265,27:FORT=0TO500:NEXT:SYS210
07                                        <122>
800 SYS 20480                             <189>
805 IFPEEK(56320)<>111THEN805             <85>
810 POKESI+11,0:POKESI+18,0:POKESI+4,0    <11>
820 POKE53280,0:POKE53281,0               <137>
830 PRINT"{CLEAR WHITE DOWN9 SPACE4}BESC
HLEUNIGUNG{SPACE}:                        <206>
840 PRINT"{DOWN SPACE4}ANFANGSGESCHW.{SP
ACE}:                                     <199>
850 PRINT"{DOWN SPACE7}'GEFALLENE'{SPACE
}:                                        <167>
860 PRINT"{HOME DOWN9 RIGHT21}({SPACE}'1
'{SPACE}BIS{SPACE}'6'{SPACE})             <162>
865 GETA$:IFA$<"1"ORA$>"6"THEN865         <253>
870 PRINT"{LEFT19}0.0"A$"{SPACE}SCHRITTE
{SPACE2}"                                 <60>
875 PRINT"{DOWN2 LEFT19}('1'{SPACE}BIS{S
PACE}'9')                                 <209>
880 GETB$:IFB$<"1"ORB$>"9"THEN880         <16>
885 PRINT"{LEFT19}0."B$"{SPACE}SCHRITTE{
```

SCHOTTER

```
SPACE3}"                                          <85>
890 PRINT"{DOWN2 LEFT19}('1'{SPACE}BIS{S
PACE}'9')                                         <224>
895 GETC$:IFC$<"1"ORC$>"9"THEN895                 <183>
900 PRINT"{LEFT19}"C$"{SPACE}SIND{SPACE}
ERLAUBT                                           <169>
910 PRINT"{BLACK DOWN4 SPACE8}ALLES{SPAC
E}RICHTIG,{SPACE}DANN{SPACE}TASTE":SYS21
007:POKE198,0                                     <62>
920 GETD$:IFD$=""THEN920                          <124>
930 IFD$="N"THEN820                               <140>
935 BE=VAL (A$)/100                               <129>
940 AG=VAL (B$)/10                                <79>
945 GF=VAL (C$)                                   <79>
1005 POKEV+1,156:POKEV+23,1:POKEV+0,255:
POKE49152+6,255                                   <49>
1010 POKEV+39,15:POKE2040,255:POKEV+2,14
6:POKEV+3,174:POKEV+29,2                          <38>
1015 POKEV+40,1:POKE2041,254:POKE2042,25
3:POKEV+41,7                                      <171>
1020 POKESI+13,0:POKESI+12,9:POKESI+24,3
1:POKESI+23,241:POKESI+8,100                      <60>
1021 POKESI+6,240:POKESI+4,33:POKESI+22,
60                                                <96>
1055 SYS21057:POKEV+21,7                          <115>
1100 POKEV+4,INT(RND(1)*245)+10:POKEV+30
,0:POKEV+5,0:POKEV+21,7:AG=AG+BE                  <64>
1105 IFPEEK(V+16)=5THENPOKEV+16,1                 <74>
1106 IFPEEK(V+16)=4THENPOKEV+16,0                 <47>
1110 FORT=0TO168STEPAG:POKEV+5,T                  <166>
1115 IFT>160THENIF(PEEK(V+30)AND6)=6THEN
1210                                              <238>
1120 POKESI+1,255-T:NEXT:IFAG>2.9THENAG=
.9                                                <15>
1200 POKESI+11,0:POKESI+11,129:FA=FA+1:I
FFA=GFTHEN1220                                    <221>
1205 POKEV+21,3:PRINT"{HOME}"TAB(21)FA:G
OTO1100                                           <201>
1210 POKESI+11,0:POKESI+11,17:POKEV+21,3
:PO=PO+11                                         <248>
1215 PRINT"{HOME}"TAB(33)PO:GOTO1100              <58>
1220 POKEV+21,0:PRINT"{CLEAR}":POKE49160
,1:FORT=0TO100:NEXT:POKEV+22,200                  <13>
1225 PRINT"{CLEAR WHITE DOWN11}AME{SPACE
}OVER{SPACE}GAME{SPACE}OVER{SPACE}GAME{S
PACE}OVER{SPACE}GAME{SPACE}OVER{SPACE}G"          <205>
1230 PRINT"{BLACK DOWN5 SPACE12}IHR{SPAC
E}SCORE{SPACE}:{SPACE}"PO:SYS21007:POKE1
98,0:POKESI+4,129                                 <10>
1235 GETA$                                        <102>
1240 POKESI+1,INT(RND(1)*255):POKESI+22,
INT(RND(1)*100)                                   <58>
1250 IFA$=""THEN1235                              <189>
1255 POKESI+4,0:PRINT"{CLEAR}":POKEV+22,
200:POKEV+16,0:GOTO10                             <243>
  ENDE DES LISTINGS
```

Aktion fehlerhaftes Listing ... Aktion fehlerhaftes Listing ...

Liebe Fans,

teilnehmen an unserer Aktion „fehlerhaftes Listing" können wirklich nur die eingesandten Coupons.

Ein Hinweis: Das Listing ist nur dann fehlerhaft, wenn das Programm sich aufhängt bzw. eine ERROR-Meldung ausgibt (richtige Programmiereingabe ist natürlich Voraussetzung).

Im letzten Heft hatte sich bei „Brückenbau" und bei „Prodata" jeweils ein kleiner Fehler eingeschlichen. Die fünf glücklichen Gewinner belohnen wir mit je 100 Mark!

„Brückenbau": Roland Möller, Horgenzell · Thomas Roedern, München · Gervin Schulze, Bad Harzburg · Michael Wirtz, Hürtgenwald · „Prodata": Mario Guerrero, Recklinghausen.

Nachweis-Coupon: Computronic (an den TRONIC-Verlag, Postfach, 3444 Wehretal 1)

Kennwort: ★ Fehlerhaftes Listing? ★ Fehlerhaftes Listing? ★ Fehlerhaftes Listing?

Name/Vorname: ____________ Straße, Nr.: ____________

PLZ/Ort: ____________ Datum, Unterschrift ____________

Ich habe folgenden Fehler in einem Listing entdeckt:

Programmname ____________ Seite ____________ Listing-Zeile ____________

richtig ist: ____________

Miner, der Fassadenstreicher

Wer hat nicht auch schon einmal den Fassadenstreichern, die in schwindelnder Höhe, auf wackeligen Gerüsten das Äußere von „alten Gemäuern" verschönen, bewundernd bei der Arbeit zugesehen. Mit dem Spiel „Miner" haben Sie nun selbst die Gelegenheit den produktiven und gefährlichen Beruf eines Fassadenstreichers auszuüben.

(Siegfried Görk)

TOP-Programm für VC-20 + 3K



„Nee, nee, Miner – bleib' lieber beim Fassadenstreichen. Das liegt Dir mehr!"

Miner in Action

Miner versucht „malerisch" tätig zu werden, ohne dabei abzustürzen oder von seinen Gegner eliminiert zu werden. Um sich am Leben zu erhalten stehen ihm einige Hilfsmittel zur Verfügung. Mit einem Hammer kann er sich für eine Weile seine Gegner vom Leibe halten. Eine Kanone bringt ihn in die Lage, sich auf eine höhere Ebene schießen zu können. Wenn er zusätzlich etwas Sprengstoff sammelt, erhöht sich zudem noch seine Sprengkraft. Um zum gegenüberliegenden Gerüstteil zu gelangen, muß er entweder die Leitern emporsteigen oder sich einen Weg mit den Transportplatten bauen.
Stürzt Miner zu tief oder wird er von einem seiner Gegner berührt, verliert er eines seiner Leben.

Zur Eingabe:

Das Programm besteht aus einem BASIC- und einem Maschinensprachteil. Um nun alle Teile in einem Programm zu vereinen, sind mehrere Schritte notwendig:
Das Maschinenprogramm ist in DATA-Zeilen abgelegt und des Speichers wegen in zwei Teile zerlegt (MINDAT1 und MINDAT2). Diese Teile müssen nacheinander gestartet werden (möglichst vorher sichern!).
Dann muß das BASIC-Programm (MINBAS) eingegeben werden. Läuft das Programm perfekt, müssen die Teile noch verbunden werden.
Dazu sind folgende Befehle einzugeben:
POKE 46,21:POKE 45,240
und
POKE 46,29:POKE 45,240.
Dann ist das Ganze sofort abzuspeichern.

MINER – Teil eins

```
          LISTING MIT CHECKSUMMEN (C V1.0)

10 POKE54,23:POKE56,23:CLR                       <237>
100 FORI=5888TO6783STEP16:ZE=0:FORJ=ITOI
+15:READW:POKEJ,W:ZE=ZE+W:NEXT                   <244>
110 READW:IFW=ZETHENNEXT:END                     <107>
120 PRINT"ERROR{SPACE}IN"1000+(I-5888)/1
.6                                               <133>
1000 DATA134,168,132,169,32,115,0,32,158
,205,32,191,209,165,101,166, 2009                <60>
1010 DATA168,164,169,96,134,168,132,169,
168,32,162,211,32,115,0,32, 1952                 <244>
1020 DATA139,208,132,74,133,73,32,208,21
9,166,168,164,169,96,134,168, 2283               <68>
1030 DATA132,169,168,16,4,169,255,208,2,
169,0,32,145,211,76,28, 1784                     <210>
1040 DATA23,32,32,32,32,32,68,68,68,0,10
1,23,36,24,131,34, 736                           <215>
1050 DATA165,2,32,46,23,165,1,32,46,23,1
65,0,76,46,23,234, 1079                          <145>
1060 DATA32,0,23,133,248,32,0,23,133,247
,32,0,23,133,251,32, 1342                        <152>
1070 DATA0,26,32,128,23,138,32,20,23,32,
240,23,138,76,20,23, 974                         <86>
1080 DATA173,62,3,201,8,144,60,173,60,3,
201,8,176,12,144,51, 1479                        <189>
1090 DATA41,6,24,105,2,205,60,3,144,41,1
65,248,41,7,240,3, 1335                          <119>
1100 DATA162,0,96,162,0,173,60,3,201,8,2
08,2,162,2,173,61, 1473                          <202>
1110 DATA3,201,8,208,1,232,173,60,3,201,
11,144,5,201,16,176, 1643                        <114>
1120 DATA1,170,96,234,162,0,160,0,165,24
8,74,32,96,24,176,18, 1656                       <8>
1130 DATA200,200,200,232,196,252,208,240
,76,154,23,165,249,217,66,3, 2681                <79>
1140 DATA208,238,88,234,234,234,134,254,
162,4,96,162,0,96,141,116, 2401                  <56>
1150 DATA162,0,165,247,41,7,208,243,165,
248,41,7,240,7,173,63, 2017                      <177>
1160 DATA3,201,9,240,7,173,61,3,201,9,20
8,2,162,1,234,234, 1748                          <21>
1170 DATA234,165,248,41,7,208,10,173,61,
3,201,10,208,3,162,2, 1736                       <33>
1180 DATA96,240,9,173,63,3,41,31,201,23,
176,9,173,61,3,41, 1343                          <233>
1190 DATA31,201,23,144,2,162,3,224,0,240
,1,96,165,247,41,7, 1587                         <44>
1200 DATA208,249,165,168,201,1,208,243,1
62,1,96,0,76,84,0,0, 1862                        <113>
1210 DATA0,0,0,83,0,130,0,0,0,0,83,80,14
1,114,144,0, 775                                 <7>
1220 DATA165,249,217,66,3,240,2,24,96,16
5,248,74,133,166,56,249, 2153                    <65>
1230 DATA65,3,16,5,201,254,144,240,96,20
1,4,176,234,56,96,127, 1918                      <246>
1240 DATA165,247,41,7,208,0,165,248,41,7
,240,7,173,63,3,201, 1816                        <171>
1250 DATA9,240,11,173,61,3,201,9,240,4,1
62,0,240,2,162,1, 1518                           <157>
1260 DATA131,82,92,0,1,0,17,131,64,0,0,0
,131,64,0,0, 713                                 <123>
1270 DATA0,131,64,0,0,0,131,64,0,0,0,131
,64,0,0,0, 585                                   <41>
1280 DATA131,64,0,0,0,131,64,0,0,0,131,6
4,0,0,0,134, 719                                 <105>
1290 DATA0,0,0,0,134,0,0,0,0,134,0,0,0,0
,134,0, 402                                      <175>
1300 DATA0,0,134,0,0,0,0,134,0,0,0,0,1
34,0,0, 402                                      <113>
```

Hohe Spielgeschwindigkeit – Hervorragende Steuerung – High-Resolution-Grafik

```
1310 DATA0,0,134,0,0,0,0,134,0,0,0,0,66,
80,27,0, 441                              <136>
1320 DATA169,93,141,15,144,169,255,141,5
,144,32,95,229,169,0,133, 1934            <224>
1330 DATA252,76,80,25,30,41,7,153,0,150,
185,0,22,153,0,31, 1205                   <63>
1340 DATA41,7,153,0,151,136,208,231,185,
64,25,153,64,3,200,192, 1813              <165>
1350 DATA15,208,245,136,132,252,76,80,25
,252,24,101,252,133,248,230, 2409         <116>
1360 DATA1,44,2,1,8,2,1,68,13,255,52,8,1
,36,6,0, 498                              <49>
1370 DATA169,27,141,21,3,169,192,141,20,
3,169,255,141,36,145,169, 1801            <246>
1380 DATA240,141,37,145,76,29,26,191,201
,219,188,139,24,135,145,76, 2012          <169>
1390 DATA162,31,169,199,63,190,96,93,177
,66,216,72,224,104,2,176, 2040            <22>
1400 DATA32,0,27,160,0,177,209,201,33,17
6,1,96,41,7,168,185, 1513                 <92>
1410 DATA59,3,96,145,96,29,26,162,0,32,0
,27,152,133,252,240, 1452                 <27>
1420 DATA76,0,27,160,0,145,209,138,145,2
43,96,169,18,56,229,252, 1963             <158>
1430 DATA32,0,27,160,0,177,243,41,7,96,2
4,101,252,133,248,230, 1771               <16>
1440 DATA169,168,133,247,169,80,133,248,
169,1,133,251,76,0,25,97, 2099            <53>
1450 DATA208,244,234,234,234,96,23,28,16
5,17,4,111,102,108,108,96, 2012           <144>
1460 DATA160,0,169,32,145,209,200,145,20
9,96,188,139,24,135,145,76, 2072          <195>
1470 DATA162,31,169,199,63,190,96,93,177
,66,216,72,224,104,2,176, 2040            <102>
1480 DATA120,166,249,164,250,32,66,27,16
2,3,188,165,27,138,24,105, 1886           <69>
1490 DATA33,209,209,208,5,189,60,3,145,2
09,202,16,237,165,247,74, 2211            <46>
1500 DATA74,74,133,249,170,165,248,74,74
,74,168,133,250,32,66,27, 2011            <71>
1510 DATA162,31,169,0,157,8,29,202,16,25
0,165,247,41,7,170,165, 1819              <64>
1520 DATA251,10,10,10,168,169,8,133,97,1
85,0,28,157,8,29,200, 1463                <17>
1530 DATA232,198,97,208,244,32,169,27,16
2,3,188,165,27,177,209,157, 2295          <25>
1540 DATA60,3,201,32,240,6,32,127,27,24,
144,4,169,1,145,243, 1458                 <241>
1550 DATA138,24,105,33,145,209,202,16,22
5,88,96,177,66,216,72,224, 2036           <148>
  ENDE DES LISTINGS
```

MINER – Teil zwei

```
      LISTING MIT CHECKSUMMEN (C V1.0)

100 FORI=6784TO7679STEP16:ZE=0:FORJ=ITOI
+15:READW:POKEJ,W:ZE=ZE+W:NEXT            <19>
110 READW:IFW=ZETHENNEXT:END              <107>
120 PRINT"ERROR{SPACE}IN"(I-6784)/1.6+10
00                                        <101>
1000 DATA166,100,165,101,74,74,168,32,66
,27,32,224,25,165,101,24, 1544            <191>
1010 DATA101,102,42,201,168,176,19,74,13
3,101,41,3,208,19,160,21, 1569            <20>
1020 DATA176,2,160,24,177,209,201,32,208
,7,169,0,56,229,102,133, 1885             <51>
1030 DATA102,165,101,74,74,168,32,66,27,
160,0,165,101,41,3,10, 1289               <213>
1040 DATA145,209,24,105,1,200,145,209,16
5,253,145,243,136,145,243,96, 2464        <86>
1050 DATA169,0,133,103,164,103,196,252,1
44,1,96,162,3,185,64,3, 1778              <245>
1060 DATA149,99,200,202,208,247,32,128,2
6,162,3,164,103,181,99,153, 2156          <116>
1070 DATA64,3,200,202,208,247,132,103,76
,212,26,34,140,147,168,100, 2062          <119>
1080 DATA234,234,169,127,141,34,145,173,
32,145,41,128,133,1,169,255, 2161         <76>
1090 DATA141,34,145,173,31,145,72,41,28,
5,1,133,1,104,41,32, 1127                 <159>
1100 DATA133,0,165,1,41,4,208,2,202,234,
165,1,41,8,208,2, 1415                    <5>
1110 DATA232,234,165,1,41,16,208,2,136,9
6,165,1,41,128,208,1, 1675                <34>
1120 DATA200,96,72,169,15,133,210,138,10
,133,209,10,24,101,209,133, 1862          <91>
1130 DATA209,138,10,10,10,10,38,210,24,1
01,209,133,209,169,0,101, 1581            <97>
1140 DATA210,133,210,165,209,133,243,24,
165,210,105,120,133,244,152,24, 2480      <117>
1150 DATA101,209,133,209,133,243,144,4,2
30,210,230,244,104,96,145,234, 2669       <228>
1160 DATA134,97,132,98,10,10,10,168,138,
10,10,10,170,169,8,133, 1307              <100>
1170 DATA99,185,0,28,29,8,29,157,8,29,20
0,232,198,99,208,241, 1750                <130>
1180 DATA166,97,164,98,96,0,22,1,23,165,
248,41,7,208,1,96, 1433                   <97>
1190 DATA168,162,15,94,8,29,126,24,29,20
2,16,247,136,208,242,96, 1802             <91>
1200 DATA162,7,181,96,72,202,208,250,165
,209,72,165,210,72,165,243, 2479          <139>
1210 DATA72,165,244,72,162,0,160,0,32,0,
27,134,1,132,2,32, 1235                   <154>
1220 DATA208,26,104,133,244,104,133,243,
104,133,210,104,133,209,162,0, 2250       <53>
1230 DATA104,149,97,232,224,7,208,248,76
,21,235,2,208,8,41,1, 1861                <71>
1240 DATA126,153,255,129,255,126,228,7,0
,0,0,0,0,0,0,0, 1279                      <173>
1250 DATA31,38,63,32,48,31,57,1,128,64,1
92,64,192,128,0,192, 1261                 <8>
1260 DATA7,9,15,8,15,7,2,14,224,144,240,
16,240,224,112,0, 1277                    <221>
1270 DATA1,2,3,3,3,1,0,3,248,100,252,252
,252,248,156,128, 1652                    <5>
1280 DATA126,66,126,66,126,66,126,66,255
,85,136,85,34,85,255,0, 1703              <6>
1290 DATA255,0,0,0,0,0,0,0,0,0,56,120,
16,16,16, 479                             <193>
1300 DATA126,94,94,94,223,191,191,126,0,
0,0,0,0,60,36,60, 1295                    <164>
1310 DATA0,0,0,0,0,255,153,255,0,0,60,36
,60,255,153,255, 1482                     <47>
1320 DATA56,84,84,124,238,56,108,255,48,
104,104,124,96,56,80,88, 1705             <170>
1330 DATA48,104,104,124,96,56,32,48,24,4
4,44,124,12,56,20,52, 988                 <101>
1340 DATA24,44,44,124,12,56,8,24,56,124,
124,124,254,184,44,96, 1342               <81>
1350 DATA56,124,124,124,254,58,104,12,25
5,127,63,31,15,7,3,1, 1358                <92>
1360 DATA128,192,224,240,248,252,254,255
,255,255,255,255,255,255,255,255, 3833    <89>
1370 DATA1,3,7,15,31,63,127,255,255,254,
252,248,240,224,192,128, 2295             <0>
1380 DATA120,252,192,120,13,141,253,248,
0,0,0,227,182,134,182,227, 2291           <154>
1390 DATA0,0,0,148,217,217,217,152,0,0,0
,224,183,240,135,224, 1957                <89>
1400 DATA0,0,0,0,0,0,0,3,6,6,7,6,3,2,3
, 36                                      <14>
1410 DATA0,0,0,0,0,0,0,0,128,128,192,0
,128,0,0, 576                             <70>
1420 DATA0,0,0,0,0,0,0,0,98,100,8,16,3
8,70,0, 330                               <176>
1430 DATA48,72,72,48,74,68,58,0,4,8,16,0
,0,0,0,0, 468                             <102>
1440 DATA4,8,16,16,16,8,4,0,32,16,8,8,8,
16,32,0, 192                              <12>
1450 DATA8,42,28,62,28,42,8,0,0,8,8,62,8
,8,0,0, 312                               <151>
```

```
1460 DATA0,0,0,0,0,8,8,16,0,0,0,126,0,0,
0,0, 158                                   <208>
1470 DATA0,0,0,0,0,24,24,0,0,2,4,8,16,32
,64,0, 174                                 <54>
1480 DATA126,126,102,102,102,102,126,126
,56,56,24,24,24,24,24,24, 1168             <108>
1490 DATA126,126,6,126,126,96,126,126,12
6,126,6,126,126,6,126,126, 1626            <77>
1500 DATA96,108,108,126,126,12,12,12,126
,126,96,126,126,6,126,126, 1458            <92>
1510 DATA126,126,96,126,126,102,126,126,
126,126,6,6,6,6,6,6, 1242                  <4>
1520 DATA126,126,102,126,126,102,126,126
,126,126,102,126,126,6,126,126, 1824       <202>
1530 DATA0,0,8,0,0,8,0,0,0,0,8,0,0,8,8,1
6, 56                                      <43>
1540 DATA14,24,48,96,48,24,14,0,0,0,126,
0,126,0,0,0, 520                           <255>
1550 DATA112,24,12,6,12,24,90,69,73,67,7
2,69,78,68,69,70, 915                      <135>
  ENDE DES LISTINGS
```

MINER ist top! – Teil drei

LISTING MIT CHECKSUMMEN (C V1.0)

```
5 POKE54,23:POKE56,23:CLR:GOTO1000         <83>
10 PRINT"{HOME RIGHT4 GREEN}"MID$(STR$(S
C),2):RETURN                               <68>
1000 SYS256*25:JO=5968:SP=5984:POKE36879
,8:FF=38400-7680:SU=0:MA=3:LV=0            <202>
1010 XM=20:YM=168:SYSSP,XM,YM,18,A,A:R=1
7                                          <196>
1500 POKE37157,255-40*LV                   <194>
2000 DATA16,3,0,3,3,4,5,3,13,4,4,10,2,5,
17,5,7,0,6,8,14,5,9,8,4,10,5,1,12,5,2,12
,8,4                                       <181>
2001 DATA14,0,4,14,12,10,16,2,4,18,17,3,
20,13,5                                    <74>
2002 DATA2,3,3,4,12,7,10                   <176>
2003 DATA4,2,5,2,15,13,2,17,18             <111>
2004 DATA4,3,11,6,1,11,9,19,15             <131>
2005 DATA0                                 <191>
2006 DATA0                                 <192>
2010 DATA16,4,19,3,5,0,2,5,3,8,5,14,4,7,
6,4,8,14,2,8,17,5,9,0,4,11,3,2,11,6,6,13
,11,4                                      <242>
2011 DATA14,0,4,15,13,4,18,10,4,18,17,5,
19,4,3                                     <86>
2012 DATA3,5,2,4,8,16,7,11,5,8             <146>
2013 DATA4,4,15,6,7,12,12,17,20            <113>
2014 DATA4,4,10,7,15,10,8,17,11            <134>
2015 DATA0                                 <201>
2016 DATA0                                 <202>
2020 DATA15,3,6,1,3,8,1,4,11,4,7,5,4,7,1
2,7,7,20,2,11,0,5,11,8,2,11,11,2,11,18,3   <107>
2021 DATA16,8,4,16,15,2,16,20,2,18,2,4,2
0,8,3                                      <230>
2022 DATA3,3,7,4,7,19,4,11,10,5            <97>
2023 DATA4,6,17,10,2,15,15,17,3            <165>
2024 DATA4,3,13,10,9,15,8,21,14            <183>
2025 DATA3,11,13,16,12,16,17               <95>
2026 DATA0                                 <212>
2030 DATA14,4,0,7,5,9,2,5,14,8,8,3,5,8,1
9,3,9,12,4,10,9,2,12,0,5,12,16,3,15,0,6,
15,10,3                                    <251>
2031 DATA15,19,3,18,0,7,18,16,6            <89>
2032 DATA3,12,5,3,15,6,3,18,7,4            <248>
2033 DATA3,4,16,7,5,11,17                  <133>
2034 DATA4,9,9,17,20,21,2,21,12            <34>
2035 DATA1,15,13                           <39>
2036 DATA1,"{HOME DOWN11 GREEN}MMM{DOWN3
 LEFT3}NNN{DOWN3 LEFT3}OOO{DOWN4 RIGHT17
 WHITE}L{HOME}"                            <165>
2040 DATA13,3,11,8,6,0,1,6,2,8,6,12,6,10
,11,3,12,6,12,13,0,2,16,0,4,16,5,6,16,14
,7                                         <228>
2041 DATA18,10,5,3,20,2,8,19,3             <95>
2042 DATA3,6,1,7,16,4,6,3,19,5             <190>
2043 DATA5,2,15,5,2,11,11,15,5,15,17       <84>
2044 DATA0                                 <230>
2045 DATA0                                 <231>
2046 DATA0                                 <232>
2050 DATA16,3,7,3,3,11,2,5,4,6,5,11,3,8,
8,3,9,13,7,9,21,1,11,1,4,13,6,2,13,13,4,
15,0,4                                     <6>
2051 DATA15,9,2,17,6,3,17,13,2,19,2,2,20
,12,4                                      <160>
2052 DATA2,3,10,5,9,20,13                  <62>
2054 DATA4,2,8,8,17,10,3,19,14             <175>
2055 DATA3,7,9,14,9,21,8                   <161>
2056 DATA0                                 <243>
2057 DATA0                                 <244>
2060 DATA13,4,6,5,3,13,3,6,0,4,9,8,7,9,1
6,4,14,0,8,14,14,8,16,10,2,17,0,3,19,5,5   <117>
2061 DATA19,12,5,5,16,5,10,4,2             <254>
2062 DATA2,6,4,4,9,15,5                    <107>
2063 DATA6,5,2,4,18,8,11,8,17,18,7,18,14   <14>
2064 DATA3,3,8,15,10,15,11                 <161>
2065 DATA0                                 <252>
2066 DATA0                                 <253>
2070 KL=0:XM=20:YM=168:RESTORE:IFSU=0THE
N2080                                      <156>
2071 FORI=1TOSU:FORJ=0TO1:READZ:FORK=1TO
Z:READX,Y,L:NEXT                           <228>
2072 NEXT:FORJ=0TO2:READZ:IFZTHENFORK=1T
OZ:READX,Y:NEXT                            <126>
2073 NEXT:READZ:IFZTHENFORJ=1TOZ:READA$:
NEXT                                       <234>
2074 NEXT                                  <164>
2080 FORI=0TO1000*(5-SU):NEXT:POKE252,0:
PRINT"{CLEAR}":IFSU=7THENSU=0:LV=LV+1:GO
TO1500                                     <113>
2081 FM=22:READZ:FORJ=1TOZ:READY,X,L:P=7
680+22*Y+X:FORI=PTOP+L-1:POKEI,9:POKEFF+
I,7:NEXT                                   <55>
2082 FM=FM+L:NEXT                          <43>
2083 FORI=7680+484TOI+22:POKEI,9:POKEFF+
I,7:NEXT                                   <172>
2084 READZ:FORJ=1TOZ:READY,X,L:P=7680+22
*Y+X:FORI=PTOP+22*L-1STEP22:POKEI,8:POKE
FF+I,1                                     <179>
2085 NEXTI,J                               <237>
2086 READZ:FORI=0TOZ*3-1STEP3:READY,X:PO
KE832+I,1:POKE834+I,Y:POKE833+I,X*4:NEXT   <54>
2087 POKE252,Z*3:POKE253,1                 <69>
2088 READZ:IFZTHENFORI=1TOZ:READY,X:POKE
7680+22*Y+X,11:POKE38400+22*Y+X,3:NEXT     <178>
2089 READZ:IFZTHENFORI=1TOZ:READY,X:POKE
7680+22*Y+X,10:NEXT                        <237>
2090 READZ:IFZTHENFORI=1TOZ:READA$:PRINT
A$;:NEXT                                   <85>
2091 FORI=1TOMA:POKE7692+I,16:IFI<8THENN
EXT                                        <1>
2099 PRINT"{HOME CYAN}\]^_"                <131>
2998 SO=36876:RA=36877                     <110>
2999 POKE36877,0:POKE36876,0:POKE36878,1
5                                          <128>
3000 SYSSP,XM,YM,R+Z,L,B:SYSJO,XR,YR,S:X
A=XR*2:IFS=0THEN4000                       <30>
3005 P=22*PEEK(249)+PEEK(250)+38422:IFB<
>1OR(PEEK(P)AND7)<>7THEN3008               <12>
3006 POKEP,4:SC=SC+1:GOSUB10:FM=FM-1:IFF
M>0THEN3008                                <65>
3007 MA=MA+1:SU=SU+1:SC=SC+SU*10^LV:GOTO
2070                                       <108>
3008 IFL>3GOTO4100                         <57>
3009 GZ=GZ-1:IFGZ=0THENPOKE253,1           <129>
3010 IF(XRANDL<>3ANDB=1ANDXA+XM>=0ANDXA+
XM<170)=0THEN3020                          <251>
3011 POKERA,250+3*Z:R=18-XR:Z=Z+1AND1:XM
=XM+XA:POKERA,0:GOTO3000                   <233>
3020 IFYRANDL=3THENYM=YM+YR*2:Z=Z+1AND1:
```



```
GOTO3000                                       <104>
3030 IFYR=-1ANDL=2THENYM=YM-2:R=21:Z=Z+1
AND1:GOTO3000                                  <240>
3040 IFYR=1ANDL=1THENYM=YM+2:R=21:Z=Z+1A
ND1:GOTO3000                                   <118>
3050 IFYR=-1ANDL=1AND(YMAND7)THENYM=YM-2
:Z=Z+1AND1:GOTO3000                            <209>
3055 FA=0                                      <152>
3057 IFB=2THEN3100                             <229>
3060 IFB=0ANDL=0THENFA=FA+1:YM=YM+2:SYSS
P,XM,YM,R,L,B:POKE36876,240-2*FA:GOTO306
0                                              <88>
3065 POKE36876,0:IFFA>16THEN4300               <179>
3070 IFB=2THEN3100                             <242>
3099 GOTO3000                                  <86>
3100 XX=-1:P=PEEK(249)*22+PEEK(250)+7680
:IFPEEK(P+21)<>32THENXX=1                      <102>
3110 POKEP+22,32                               <38>
3115 XM=XM+XX*2:SYSSP,XM,YM,16,L,B             <79>
3120 IF(XMAND7)>0ORPEEK(PEEK(249)*22+PEE
K(250)+7702+XX)<>9ANDPEEK(831)<>9THEN311
5                                              <114>
3130 POKEPEEK(249)*22+PEEK(250)+7702,10:
XM=XM+XX*2:SYSSP,XM,YM,R+Z,L,B:GOTO3010        <167>
3999 GOTO3000                                  <221>
4000 YM=YM-2                                   <253>
4001 FORA=-4TO20:FORYM=YMTOYM+A*2STEPSGN
(A)*2:SYSSP,XM,YM,R+Z,L,B                      <189>
4005 IFB=0THENNEXT:POKE36876,230-2*ABS(A
):XM=XM-XA*((XM+XAAND255)<170):NEXT            <232>
4006 POKE36876,0:IFA>6THEN4300                 <122>
4010 XA=0:GOTO3005                             <104>
4100 IFL=4ANDGZ<=0THEN4300                     <6>
4105 IFL=4ANDGZ>0THEN4200                      <15>
4110 IFL=11THENPOKESO,200:POKE828,32:POK
E253,4:GZ=50:SC=SC+10:GOSUB10:POKESO,0:G
OTO3010                                        <32>
4120 IFL>11ANDL<16THEN4400                     <99>
4200 POKESO,140:ZA=PEEK(252):P=832+3*PEE
K(254):POKE252,0:PE=ZA+829                     <176>
4205 PZ=7680+22*PEEK(P+2)+PEEK(P+1)/4:PO
KEPZ,32:POKEPZ+1,32                            <221>
4210 FORI=0TO2:POKEP+I,PEEK(PE+I):NEXT         <115>
4220 POKE252,ZA-3:POKESO,0:GOTO3010            <83>
4300 FORI=15TO0STEP-1:POKE36878,I:POKE36
877,200+I*2:SYSSP,XM,8+YM-I/2,R+Z,A,A:NE
XT                                             <6>
4305 MA=MA-1:IFMATHEN2070                      <79>
4310 WAIT0,32:WAIT0,32,32:WAIT0,32             <142>
4320 RUN                                       <122>
4400 IFL>12THENPOKESO,220:POKE828,32:KL=
KL+L-12:POKESO,0:GOTO3010                      <237>
4410 SYSSP,XM,YM,12,L,B:POKE828,32:P=768
0+22*PEEK(249)+PEEK(250)                       <205>
4415 SYSSP,XM,YM,12,L,B                        <174>
4420 SYSJO,X,Y,S:X=X*2:IF(XM+XAND255)<16
8THENXM=XM+X                                   <254>
4430 IFSTHEN4415                               <242>
4435 POKERA,220                                <136>
4440 FORI=12TO1STEP-1:YM=YM-1.2^I*(KL/1.
5+.5):POKE36878,I*1.25                         <71>
4445 IFYM>0THENSYSSP,XM,YM,R,L,B:NEXT:GO
TO4460                                         <2>
4450 POKERA,0:POKE36878,15:KL=0:GOTO4300       <179>
4460 POKERA,0:POKE36878,15:KL=0:YM=YMAND
254:SYSSP,XM,YM,R,L,B:POKEP,12:GOTO3055        <251>
  ENDE DES LISTINGS
```

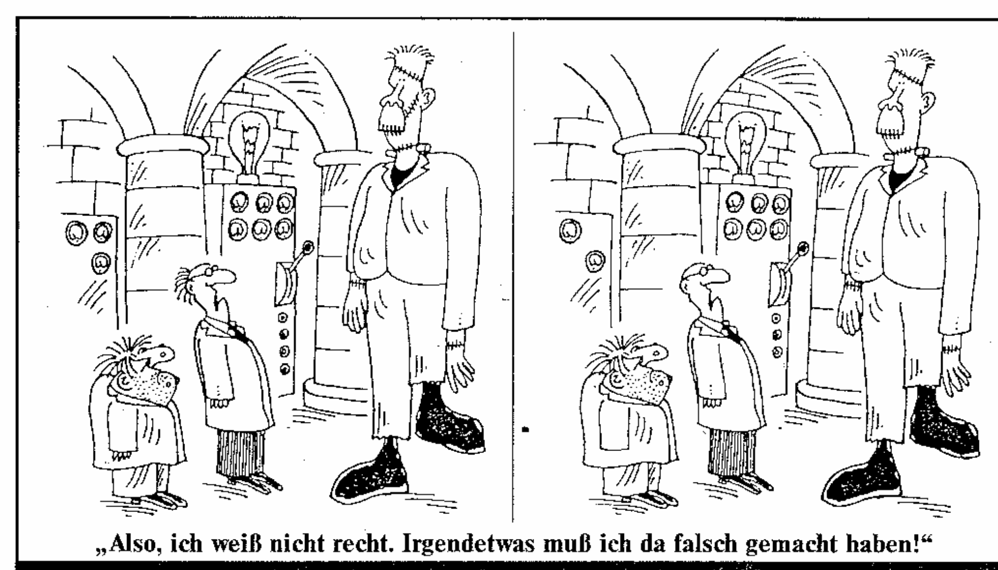
——— Okay, Miner, dann mal ran an die Arbeit! Aber fall nicht vom Gerüst! ———

WETTBEWERB: Auch diesmal wieder viele tolle Preise zu gewinnen!

Vorsicht, Leute, aufgepaßt! Wie uns aus gutinformierten Kreisen mitgeteilt wurde, versucht ein raffinierter Betrüger, gefälschte Kunstwerke „an den Mann zu bringen". Nach harten Anstrengungen ist es uns endlich gelungen, ein solches Machwerk „ergattern" und für Sie abdrucken zu können.

Wir hoffen, Sie dadurch vor einem möglichen „Reinfall" bewahren zu können. Wenn Sie die nebenstehende Abbildung genau betrachten, werden Sie feststellen, daß sich das rechte Bild vom linken in fünf Punkten unterscheidet. Offenbar hat es unserem „Künstler" an der Zeit gefehlt, um eine vollständige Imitation des Originals anfertigen zu können.

Ihre kriminalistische Mitarbeit belohnen wir mit drei Datenträgern für den ersten, zwei Datenträgern für den zweiten und einem Datenträger für den dritten Platz, nach freier Wahl aus unserem Software-Katalog. Wenn Sie also die fünf Fehler gefunden haben, markieren Sie die Abweichungen auf der Abbildung, schneiden diese aus, kleben sie auf eine Postkarte und „spurten" zum nächsten Briefkasten. Einsenden an: **TRONIC-Verlag, Postfach 41, Wehretal 1, Kennwort: „Frankenstein". PS. Absender nicht vergessen und Datenträger-Wunsch angeben! (bez.)**



„Also, ich weiß nicht recht. Irgendetwas muß ich da falsch gemacht haben!"

Auf den Spuren von Fernando Cortez im heißen und staubigen Mexiko!

„Inkaschatz“

für den VC-20 in der Grundversion

Wer möchte nicht auch einmal auf den Spuren von Fernando Cortez wandeln und in Mexiko zahlreiche Abenteuer bestehen und die sagenumwobenen Schätze der Inkas bergen. Doch bevor Fernando an die Schätze herankommt, hat er zahlreiche Gefahren zu überstehen. ***(Siegfried Görk)***

Das Spiel besteht aus drei Teilen:
1. Zeichensatzumdefinierung
2. Maschinencode
3. Spiel

Teil 2 und 3 werden automatisch nachgeladen, gespielt wird mit dem Joystick. Bitte unbedingt die Stoptaste der Datasette drücken!

Am oberen Bildschirmrand befinden sich vier Anzeigen:
1. Score = Deine Punktzahl
2. TI = Deine Zeit
3. LI = Deine Leben
4. LE = Welcher Level?

Fernando steht zu Beginn des Spiels in der Mitte des Bildes. Seine Aufgabe ist es die sagenumwobenen Schätze, die aus Ringen, Schwertern, Diamanten, Goldbarren, Kelchen, Kronen und Leuchtern bestehen, einzusammeln. Doch auf dem Weg zu diesen Schätzen wird er von Hunden, Schlangen und Adlern bedroht.
Durch Drücken des Feuerknopfes kann Fernando - allerdings nur wenn er vorher stehen geblieben ist - seinen Gegnern springend ausweichen. Wenn ihn ein Gegner trotzdem berührt oder er in eine Fallgrube mit nach oben gerichteten Spitzen fällt, verliert er eines seiner Leben.



„Gold hin, Gold her. Lieber -ne hübsche Maid in der Heimat als ein oller Inka in Mexiko!“

Und nun auf ans Werk! – Die Inkaschätze warten auf ihre Bergung!

Variablenliste:

SC =	Punktzähler
LI =	Anzahl der Leben
LE =	Levelzähler
T =	Allgemeine Zählvariable
TI$ =	Zeitzähler
R =	Richtung des Männchens
M =	Position des Männchens
P =	Körperform des Männchens (z. B. Schritte)
P1 =	Körperform des Männchens (z.B. Schritte)
Z1 =	Peek Position des Männchens
Z2 =	Peek Position der Farbe des Männchens
F =	Zur Farbengebung
S =	Zur Tonerzeugung
K1 =	Joystickabfrage
K2 =	Joystickabfrage

Um zu den Inkaschätzen zu gelangen, steht Fernando nur ein bestimmter Zeitraum zur Verfügung. Nach Überschreiten der vorgegebenen Frist wird das Spiel beendet. Zu Beginn des Spiels beträgt der Zeitraum ca. 70 Sekunden. Mit ansteigendem Level verringert sich die Spielzeit, sobald man den Level erreicht hat, tauchen weitere bedrohliche Hunde und Schlangen auf.

INKA-SCHATZ – Teil eins

```
VC20-LISTING + CHECKSUMMER (VC V1.0)

10 POKE52,26:POKE56,26:CLR:GOTO100          <155>
15 ******************                        <243>
20 *                *                        <24>
25 *  AUF DER JAGD  *                        <233>
30 *                *                        <34>
35 *    NACH DEM    *                        <145>
37 *                *                        <41>
38 *   INKASCHATZ   *                        <81>
40 *                *                        <44>
45 ******************                        <17>
50 *                *                        <54>
55 * (C) 10/1985 BY *                        <117>
60 *                *                        <64>
65 *    O.WINDEL    *                        <26>
70 *                *                        <74>
75 * TEL. TEL. TEL. *                        <234>
80 *                *                        <84>
85 *  05121/510771  *                        <182>
90 *                *                        <94>
95 ******************                        <67>
100 POKE36879,8                              <64>
110 PRINT"{CLEAR DOWN7 WHITE RIGHT2}EIN{
SPACE}WENIG{SPACE}GEDULD,{RIGHT5 DOWN}DI
E{SPACE}DATA'S{SPACE}WERDEN{RIGHT5 DOWN}
EINGELESEN{SPACE}!"                          <135>
120 FORT=7168TO7583:READF:POKET,F:S=S+F:
NEXT                                         <176>
130 FORT=7584TO7679:POKET,0:NEXT             <96>
135 IFS<>36955THENPRINT"{CLEAR DOWN6 RIG
HT4}FEHLER{SPACE}IN{SPACE}DEN{RIGHT6 DOW
N}ZEICHENSATZ{SPACE}DATA'S{SPACE}!":END      <200>
140 PRINT"{CLEAR CYAN RIGHT5}AUF{SPACE}D
ER{SPACE}JAGD"SPC(12)"{DOWN}NACH{SPACE}D
EM"                                          <170>
150 PRINT"{DOWN2 YELLOW RVSON SPACE21 RI
GHT SPACE}I{SPACE}N{SPACE}K{SPACE}A{SPAC
E}S{SPACE}C{SPACE}H{SPACE}A{SPACE}T{SPAC
E}Z{SPACE RIGHT SPACE21}"                    <88>
160 PRINT"{DOWN4 RIGHT WHITE}COMPUTER{DO
WN}LAEDT{UP}TEIL{SPACE}2{UP BLACK}"          <156>
170 POKE631,131:POKE198,1:END                <239>
200 REM                                      <87>
210 REM ZEICHENSATZ                          <163>
220 REM                                      <107>
230 DATA24,36,66,66,36,24,24,0,1,2,4,72,
```

```
48,48,72,0                                          <108>
240 DATA62,62,28,28,8,8,28,0,85,85,127,2
8,8,8,28,0                                          <228>
250 DATA8,28,8,73,107,127,127,0,15,31,63
,126,252,248,240,0                                  <66>
260 DATA24,24,8,28,122,24,28,52,24,24,8,
28,122,24,60,102                                    <62>
270 DATA24,24,16,56,94,24,56,44,24,24,16
,56,94,24,60,102                                    <43>
280 DATA28,92,72,62,29,29,20,4,56,58,18,
124,184,184,40,32                                   <3>
290 DATA0,7,4,7,1,7,0,0,0,119,69,69,69,1
19,0,0                                              <91>
300 DATA0,119,84,118,100,87,0,0,0,119,34
,34,34,39,0,0                                       <196>
310 DATA0,71,66,66,66,119,0,0,0,71,68,70
,68,119,0,0                                         <18>
320 DATA0,32,32,120,104,254,255,126,32,3
2,56,104,124,255,240,126                            <200>
330 DATA0,14,113,97,14,16,16,15,24,28,2,
2,60,64,32,31                                       <184>
340 DATA0,240,120,62,127,249,224,128,0,1
5,30,124,254,159,7,1                                <104>
350 DATA8,8,62,8,8,8,28,62,129,255,129,2
55,129,255,129,255                                  <68>
360 DATA255,255,255,255,255,255,255,255,
0,0,16,0,16,0,0,0                                   <139>
370 DATA255,255,255,255,255,123,41,8,255
,255,255,255,255,235,202,66                         <76>
380 DATA252,255,248,254,248,252,255,254,
254,252,248,255,252,254,252,248                     <120>
390 DATA0,0,0,0,0,0,0,0,66,83,215,255,25
5,255,255,255                                       <48>
400 DATA0,0,0,0,0,0,0,0,16,148,222,255,2
55,255,255,255                                      <224>
410 DATA31,63,127,63,255,31,63,127,127,2
55,63,31,127,31,255,63                              <130>
420 DATA17,91,255,255,255,255,181,36,36,
181,255,255,255,255,91,17                           <197>
430 DATA1,5,15,63,15,15,6,2,0,4,22,63,31
,23,5,4                                             <12>
440 DATA64,96,240,240,252,240,160,128,32
,160,232,248,252,104,32,0                           <149>
450 DATA88,120,24,24,24,24,56,126,26,30,
24,24,24,24,28,62                                   <159>
460 DATA0,56,108,94,250,175,91,62,24,52,
60,118,91,175,254,46                                <50>
470 DATA0,0,0,0,66,83,215,255,0,0,0,0,16
,148,221,255                                        <115>
480 DATA8,20,34,127,34,20,8,0,0,34,119,3
4,34,34,34,34                                       <190>
  ENDE DES LISTINGS
```

INKA-SCHATZ – Teil Zwei

```
100 POKE36879,8                                     <64>
110 FORT=6700TO6952:READF:POKET,F:S=S+F:
NEXT                                                <86>
120 IFS<>28035THENPRINT"{CLEAR DOWN7 WHI
TE RIGHT4}DATAFEHLER{SPACE}IM{DOWN RIGHT
9}MASCHINENCODE":END                                <12>
130 PRINT"{CLEAR DOWN4 RIGHT WHITE}COMPU
TER{DOWN3}LAEDT{DOWN3}TEIL{SPACE}3{BLACK
}"CHR$(8)                                           <91>
140 POKE631,131:POKE198,1:END                       <209>
500 REM                                             <132>
510 REM MASCHINENCODE                               <237>
520 REM                                             <153>
530 DATA160,18,185,206,31,201,18,240,15,
201,19,240,26,136,192,1,240,3,24,144,237
,24,144                                             <176>
540 DATA30,169,32,153,206,31,169,19,200,
153,206,31,136,24,144,230,169,32,153,206
,31,169                                             <110>
550 DATA18,200,153,206,31,136,24,144,215
,173,225,31,201,18,240,3,24,144,10,169,3
2,141                                               <237>
560 DATA225,31,169,19,141,208,31,160,2,1
85,76,31,201,20,240,15,201,21,240,26,200
,192,17                                             <98>
570 DATA240,3,24,144,237,24,144,30,169,3
2,153,76,31,169,21,136,153,76,31,200,24,
144,230                                             <43>
580 DATA169,32,153,76,31,169,20,136,153,
76,31,200,24,144,215,173,77,31,201,20,24
0,3,24                                              <149>
590 DATA144,10,169,32,141,77,31,169,21,1
41,92,31,160,13,185,110,30,201,23,240,8,
200,192                                             <26>
600 DATA19,240,18,24,144,241,169,32,153,
110,30,169,23,136,153,110,30,200,24,144,
233,160                                             <114>
610 DATA17,185,110,30,201,22,240,8,136,1
92,11,240,18,24,144,241,169,32,153,110,3
0,169                                               <145>
620 DATA22,200,153,110,30,136,24,144,233
,173,122,30,201,23,240,10,173,128,30,201
,22,240                                             <226>
630 DATA16,24,144,23,169,32,141,122,30,1
69,22,141,122,30,24,144,10,169,32,141,12
8,30                                                <0>
640 DATA169,23,141,128,30,96                        <208>
  ENDE DES LISTINGS
```

INKA-SCHATZ – Teil Drei

```
10 POKE36879,8                                      <230>
20 POKE36869,255                                    <170>
30 POKE37154,127                                    <105>
40 POKE36878,15                                     <163>
50 LI=3:LE=1:S=36874                                <236>
60 TI$="000000"                                     <40>
70 GOSUB1000                                        <117>
80 POKE8145,18:POKE8027,20:POKE7802,22              <203>
90 IFLE>2THENPOKE8152,19:POKE8020,21                <36>
100 F=30720:M=7891:R=0                              <183>
110 Z1=32:Z2=1:P=8                                  <236>
120 K1=PEEK(37137)                                  <250>
130 K2=PEEK(37152)                                  <230>
135 IFPEEK(M+22)=51THEN700                          <228>
140 IFPEEK(M+22)=32ORPEEK(M+22)<6ORPEEK(
M+22)=50THENR=22:P=10:GOTO220                       <130>
150 IFK1=94ANDZ1=32THENGOSUB600                     <193>
160 IFK1=110THENR=-1:P=6                            <90>
170 IFK2=119THENR=1:P=8                             <200>
180 IFK1=118THENR=22:P=10                           <95>
190 IFK1=122ANDZ1=25THENR=-22:P=10                  <228>
210 IFPEEK(M+R)<44ANDPEEK(M+R)>25ANDPEEK
(M+R)<>32THENR=0                                    <187>
220 SYS6700                                         <59>
230 IFPEEK(M)<24ANDPEEK(M)>17THEN700                <4>
240 IFR=0THEN320                                    <72>
250 POKEM+F,Z2:POKEM,Z1                             <81>
260 M=M+R                                           <223>
270 Z1=PEEK(M):Z2=PEEK(M+F)                         <151>
280 IFZ1<6ORZ1=50THENGOSUB500                       <250>
290 IFZ1<24ANDZ1>17THEN700                          <43>
300 IFP1=0THENP1=1:GOTO320                          <126>
310 P1=0                                            <191>
320 POKEM+F,1:POKEM,P+P1                            <107>
330 IFR<>0THENPOKES+3,190:POKES+3,0                 <13>
340 IFVAL(RIGHT$(TI$,3))>(120-10*LE)THEN
ZE=1:GOTO700                                        <36>
350 PRINT"{HOME WHITE RVSON}"SPC(11)RIGH
T$(TI$,3)                                           <250>
360 R=0:GOTO120                                     <72>
500 FORT=215TO235STEP.5:POKES+2,T:NEXT:P
OKES+2,0                                            <60>
520 POKES+2,0                                       <9>
530 IFZ1=50THENSC=SC+10:GOTO550                     <70>
```

```
540 SC=SC+Z1+1                                   <134>
550 SC$=STR$(SC)                                 <199>
560 SC=SC+10000:SC$=STR$(SC):PRINT"{HOME
 RVSON WHITE RIGHT4}"RIGHT$(SC$,4):SC=SC
-10000                                           <62>
570 Y=Y+1:IFY=14THENLE=LE+1:Y=0:GOTO60           <152>
580 Z1=32:RETURN                                 <28>
600 IFP=6ORP=7THENSR=-1:P=7                      <199>
610 IFP=8ORP=9THENSR=1:P=9                       <184>
620 IFPEEK(M-22+SR)<>32THENP=P-1:RETURN          <168>
625 FORT=240TO250:POKES,T:NEXT:POKES,0           <71>
630 POKEM,32:POKEM+F,Z2                          <52>
640 M=M-22+SR                                    <12>
650 IFPEEK(M)<>32THENRETURN                      <23>
660 POKEM+F,1                                    <223>
670 POKEM,P                                      <149>
680 P=P-1:P1=0:R=22+SR:RETURN                    <124>
700 POKEM,32                                     <57>
710 POKEM+F,1:POKEM,24:POKES+3,225               <133>
720 FORT=15TO0STEP-(T/150):POKES+4,T:NEX
T                                                <255>
730 POKES+3,0:POKES+4,15                         <107>
740 LI=LI-1:IFLI=0ORZE=1THENPOKE7697,176
:GOTO900                                         <49>
750 POKE7697,LI+176                              <187>
760 IFPEEK(M+22)=51THENPOKEM,32:GOTO100          <54>
770 FORT=7802TO7808:POKET,32:POKET+F,4:N
                                                 <30>
780 FORT=8013TO8028:POKET,32:POKET+F,5:N
EXT                                              <249>
790 FORT=8144TO8160:POKET,32:POKET+F,3:N
EXT                                              <197>
800 GOTO80                                       <172>
900 PRINT"{HOME DOWN8 RIGHT5 RVSON}┌──────
───────┐"                                       <59>
910 PRINT"{RIGHT5 WHITE RVSON}|GAME{SPAC
E}OVER|"                                         <251>
920 PRINT"{UP3 RIGHT5 RVSON}└──────────┘"        <186>
930 IFZE=1THENPRINT"{DOWN WHITE RVSON YE
LLOW}────────────────DEINE{SPACE}Z
EIT{SPACE}IST{SPACE}ZU{SPACE}ENDE";              <119>
940 IFZE=1THENPRINT"{RVSON}______________
___________"                                     <210>
950 WAIT37151,32,32                              <24>
960 RUN                                          <77>
1000 PRINT"{CLEAR WHITE}LMN[{RVSON}0000{
RIGHT RVSOFF}OI{RVSON}OOO{RIGHT RVSOFF}P
[{RVSON RIGHT RVSOFF RIGHT}QI{RVSON RIGH
T PURPLE}────────────────{RVSOFF}"
;                                                <53>
1010 PRINT"{GREEN}.{SPACE}../{SPACE}/.{S
PACE2}/{SPACE}..{SPACE2}//{SPACE}./{RED
},{GREEN}O{RED},--{GREEN}1{RED},,{GREEN}
10{RED}-{GREEN}1{RED}-,{GREEN}OO{RED},,{
GREEN}1{RED}--,";                                <87>
1020 PRINT"{BLUE}Z\\]\]]\\]ZZ\]\\]\]\]Z^
{RIGHT9}$_{PURPLE SPACE7 WHITE}A{YELLOW}
C{BLUE}%";                                       <145>
1030 PRINT"_{WHITE}2{RIGHT2 YELLOW}@{BLU
E RIGHT5}%Z{RED}Y{BLUE}!#{RED}Y{BLUE}!#!
#!ZZ!!{SPACE}##{SPACE}!#{RED}Y{BLUE}Z]{R
ED}Y{BLUE}\]\\]\]\Z";                            <64>
1040 PRINT"ZZZ{WHITE}3{BLUE}ZZ{WHITE}3{B
LUE}ZZ{RED}Y{BLUE}^{YELLOW}D{RED}Y{RIGHT
8 BLUE}$Z\]]\\]\\{RED}Y{BLUE}\#{RED}Y{RI
GHT7 YELLOW}B{BLUE}%";                           <106>
1050 PRINT"^{RIGHT8 RED}Y{YELLOW}C{BLUE}
$Z&+{RIGHT}){RED}Y{BLUE}*{RIGHT}(Z_{RIGH
T YELLOW}E{RIGHT3 RED}Y{BLUE}!#Z#Z_{RIGH
T4 RED}Y{BLUE RIGHT3}$";                         <184>
1060 PRINT"Z{RED}Y{BLUE}!{RIGHT}!#ZZZZZZ
^{RIGHT4 RED}Y{RIGHT WHITE}A{RIGHT BLUE}
%Z{RED}Y{BLUE}]{SPACE}\]\\]\]]]&'&']'&{R
ED}Y{BLUE}Z";                                    <54>
1070 PRINT"Z{RED}Y"SPC(18)"Y{BLUE}ZZ{RED
}Y{WHITE}2{GREEN SPACE16 YELLOW}@{RED}Y{
BLUE}Z";                                         <73>
1080 PRINT"ZZ#{RIGHT}!##{RIGHT}!#{RIGHT}
!!!#!!{RED}Y{BLUE}!#ZZZZZ{WHITE}3{BLUE}Z
ZZ{WHITE}3{BLUE}ZZ{WHITE}3{BLUE}Z\]\\]{R
ED}Y{BLUE}\]]Z";                                 <242>
1090 PRINT"ZZZZZZZZZZ_{RIGHT YELLOW}D{B
LUE RIGHT3 RED}Y{YELLOW RIGHT}C{RIGHT BL
UE}%Z\]]\]\\\]\\&'&'']'&{RED}Y{BLUE}Z";          <12>
1100 PRINT"^"SPC(19)"{RED}Y{BLUE}Z_{YELL
OW}E{CYAN SPACE18 RED}Y{BLUE}ZZ!!##!###!
!#!!##!#!!Z{HOME}"                               <24>
1110 POKE38905,6:POKE8185,26                     <161>
1120 POKE7697,LI+176                             <47>
1130 POKE7701,LE+176                             <218>
1140 RETURN                                      <6>
  ENDE DES LISTINGS
```

Mit dem C-16-Computer auf den Spuren des adligen Vorfahren im

„HORROR-CASTLE"

Dieses Spiel erscheint auf den ersten Blick recht einfach, doch sobald mehr Monster auf dem Bildschirm erscheinen, erschwert sich die Spielaufgabe ungemein. Auch Sie als erfahrener Homecomputer-Spieler kommen dann mit Sicherheit ins „Rotieren". ***(S.G.)***

Ihr adeliger Urgroßvater John McNeal hat im vorigen Jahrhundert ein Jagdschloß im stürmischen und sagenumwobenen schottischen Hochland erworben. Nach John McNeals Tod werden Sie überraschend als dessen Erbe eingesetzt.
Voller Neugier und voller Erwartung machen Sie sich umgehend auf den Weg. Bei der Begutachtung des Schlosses lernen Sie gleich die Tücke des Objekts kennen. Auf der ersten Besichtigungstour befördert Sie eine raffiniert angebrachte Falltür in ein labyrinthartiges Verlies, wo schon so mancher sein Leben lassen mußte.
Doch besteht eine Chance mit dem Leben davon zu kommen. In Ihrem Besitz befindet sich ein Bauplan des Verlieses, in dem sämtliche Geheimgänge und -türen eingezeichnet sind. Mit Hilfe dieses Planes können Sie an die versteckten Notvorräte gelangen, die für Ihre Person lebensnotwendig sind (= Punktegutschrift).
Doch aufgepaßt? Bei Ihrer Nahrungssuche werden Sie von blutrünstigen Vampiren bedroht. Die Geheimgänge und Geheimtüren sind nur von Ihnen allein passierbar, also eine Möglichkeit den Vamps auszuweichen.
Punktewertung: Bonbons vertilgen – für 2 Bonbons die doppelten Punkte; für vier das dreifache; für sechs das vierfache.
Mit Hilfe der Äpfel müssen Sie versuchen die Begriffe „EXTRA" und „SPECIAL" zu vervollständigen. Diese Äpfel gilt es in diesem Augenblick zu vertilgen, in dem unten links auf dem Bildschirm einer der fehlenden Buchstaben erscheint. Wenn der Begriff „EXTRA" vervollständigt ist, erhält man ein Zusatzleben, nach Vollendung des Wortes „SPECIAL" wird ein Bonus von 17000 Punkten erteilt.
Durch Betätigen des Feuerknopfes (Joystick Port 1) können Sie sich bis zur nächsten Joystickbewegung unsichtbar machen. Achten Sie dabei auf die Help-Anzeige, Sie können nur dann verschwinden, wenn die Help-Anzeige nicht auf Null steht.
Das Spielgeschehen von „Horror-Castle" ist größten Teils in Maschinensprache geschrieben.
Das Programm startet mit RUN und lädt

John McNeal's Vermächtnis



anschließend Zeichensatz und Maschinenspracheroutine ein. Bitte zuerst die Datas der Maschinensprachroutine eingeben, danach die Datas für den Zeichensatz und schließlich das Hauptprogramm, das Sie aus dem Basic heraus saven müssen.
Zeichensatz und Maschinenspracheroutine sind aus dem Monitor mit S „zs, mc", 3400, 3f00 abzusaven.

Teil eins

```
100 REM    ALL RIGHTS RESERVED
101 REM **************************
102 REM *                        *
103 REM *    R & R PRESENTS:     *
104 REM *                        *
105 REM *   CASTLE OF HORROR     *
106 REM *                        *
107 REM **************************
108 REM *                        *
109 REM *   CHRISTIAN  RASKOB    *
110 REM *                        *
111 REM *    GARTENSTRASSE 6     *
112 REM *                        *
113 REM *   5561 GROSSLITTGEN    *
114 REM *                        *
115 REM *    TEL.:   06575/8111  *
116 REM *                        *
117 REM **************************
118 POKE55,0:POKE56,52:CLR:RESTORE:
    FORI=14336TO16127STEP16:SU=0
119 FORJ=0TO15
120 READZ:SU=SU+ZAND255
121 POKEI+J,Z
122 NEXTJ
123 READS:IFS<>SUTHEN238
124 NEXTI
125 GOTO240
```

```
126 DATA 189,251,191,170,120,142,8,255,173,8,255,142,8,255,205,8,76
127 DATA 255,208,242,174,8,255,224,247,208,3,76,167,57,224,251,208,247
128 DATA 3,76,203,57,224,254,208,3,76,239,57,224,253,208,3,76,116
129 DATA 19,58,224,191,208,3,76,146,62,96,234,234,165,208,56,101,33
130 DATA 232,144,2,230,209,133,208,96,165,208,24,229,232,176,2,198,184
131 DATA 209,133,208,96,165,210,133,208,165,211,133,209,96,165,208,133,122
132 DATA 210,165,209,133,211,96,198,209,198,209,198,209,198,209,96,234,166
133 DATA 234,234,234,234,32,84,56,160,0,173,52,3,145,208,162,0,219
134 DATA 134,232,234,32,60,56,173,53,3,145,208,162,38,134,232,234,82
```

```
135 DATA 32,60,56,173,54,3,145,208,162,0,134,232,234,32,60,56,105
136 DATA 173,55,3,145,208,32,84,56,32,102,56,173,72,3,145,208,11
137 DATA 162,0,134,232,234,32,60,56,173,73,3,145,208,162,38,134,54
138 DATA 232,234,32,60,56,173,74,3,145,208,162,0,134,232,234,32,219
139 DATA 60,56,173,75,3,145,208,96,162,4,202,189,52,3,157,60,109
140 DATA 3,189,72,3,157,68,3,189,56,3,157,52,3,189,64,3,187
141 DATA 157,72,3,224,0,208,227,32,116,56,162,4,202,189,60,3,179
142 DATA 157,52,3,189,68,3,157,72,3,224,0,208,239,96,32,84,51
143 DATA 56,165,226,141,232,0,32,72,56,32,93,56,32,84,56,160,213
144 DATA 0,177,208,141,56,3,162,0,142,232,0,32,60,56,177,208,118
145 DATA 141,57,3,162,38,142,232,0,32,60,56,177,208,141,58,3,230
146 DATA 162,0,142,232,0,32,60,56,177,208,141,59,3,32,84,56,164
147 DATA 32,102,56,177,208,141,64,3,162,0,142,232,0,32,60,56,187
148 DATA 177,208,141,65,3,162,38,142,232,0,32,60,56,177,208,141,50
149 DATA 66,3,162,0,142,232,0,32,60,56,177,208,141,67,3,32,101
150 DATA 116,56,165,229,145,220,165,230,145,222,96,32,84,56,166,224,47
151 DATA 134,232,32,72,56,160,0,177,208,133,229,166,225,134,232,32,174
152 DATA 60,56,177,208,133,230,96,162,0,134,226,162,1,134,224,162,117
153 DATA 39,134,225,162,60,160,72,142,23,57,142,160,57,142,147,57,243
154 DATA 162,57,134,220,162,59,134,222,108,76,3,162,0,134,226,162,229
155 DATA 0,134,224,162,39,134,225,162,60,160,72,140,23,57,142,160,102
156 DATA 57,140,147,57,162,56,134,220,162,58,134,222,108,76,3,162,106
157 DATA 39,134,226,162,39,134,224,162,0,134,225,162,60,160,72,140,25
158 DATA 23,57,142,160,57,140,147,57,162,56,134,220,162,57,134,222,138
159 DATA 108,76,3,162,39,134,226,162,79,134,224,162,0,134,225,162,238
160 DATA 60,160,72,142,23,57,142,160,57,142,147,57,162,58,134,220,1
161 DATA 162,59,134,222,108,76,3,173,1,255,41,3,170,224,0,208,47
162 DATA 2,160,1,224,1,208,2,160,0,224,2,208,2,160,3,224,45
163 DATA 3,208,2,160,2,204,78,3,208,2,152,170,142,78,3,224,103
164 DATA 0,208,3,76,167,57,224,1,208,3,76,203,57,224,2,208,181
165 DATA 3,76,239,57,224,3,208,3,76,19,58,96,169,137,133,231,196
166 DATA 169,58,133,232,165,229,76,150,58,165,212,133,231,165,213,133,218
167 DATA 232,165,230,76,150,58,201,32,208,3,108,231,0,201,96,208,151
168 DATA 3,108,231,0,201,69,208,3,76,46,59,201,70,208,3,76,26
169 DATA 46,59,201,71,208,3,76,46,59,201,72,208,3,76,46,59,154
170 DATA 201,73,208,3,76,52,59,201,79,208,3,76,81,59,201,77,121
171 DATA 208,3,76,117,61,201,78,208,3,76,75,59,76,138,61,32,192
172 DATA 111,61,234,96,201,65,208,3,76,118,60,201,66,208,3,76,251
173 DATA 118,60,201,67,208,3,76,118,60,201,68,208,3,76,118,60,109
174 DATA 201,78,208,3,108,231,0,201,79,208,3,108,231,0,234,76,177
175 DATA 87,59,234,234,169,231,133,231,169,60,133,232,165,229,76,228,110
176 DATA 58,165,214,133,231,165,215,133,232,165,230,76,228,58,169,1,169
177 DATA 141,79,3,96,238,80,3,165,231,201,137,208,7,169,32,133,131
178 DATA 229,108,231,0,169,32,133,230,108,231,0,238,81,3,76,55,132
179 DATA 59,238,82,3,76,55,59,168,201,73,208,3,108,231,0,201,229
180 DATA 32,208,3,108,231,0,76,66,60,234,234,234,0,174,96,3,223
181 DATA 202,202,32,181,60,202,202,202,160,0,136,200,232,232,232,232,147
182 DATA 32,220,62,232,189,0,63,153,52,3,189,64,63,153,72,3,14
183 DATA 189,128,63,153,56,3,189,192,63,153,64,3,192,3,208,219,86
184 DATA 174,96,3,189,64,55,133,210,189,80,55,133,211,189,96,55,140
185 DATA 141,78,3,169,59,141,77,3,169,192,141,76,3,76,55,58,161
186 DATA 32,139,57,169,59,133,215,169,209,133,214,76,235,59,76,20,203
187 DATA 59,32,216,56,32,14,57,76,241,59,174,96,3,232,228,216,255
188 DATA 240,6,142,96,3,76,109,59,76,58,60,32,206,59,76,136,154
189 DATA 60,174,96,3,202,202,202,202,32,208,62,202,160,0,136,200,93
190 DATA 32,220,62,232,232,232,232,232,185,52,3,157,0,63,185,56,127
191 DATA 3,157,128,63,185,64,3,157,192,63,185,72,3,157,64,63,23
192 DATA 192,3,208,219,174,96,3,165,210,157,64,55,165,211,157,80,111
```



```
193 DATA 55,173,78,3,157,96,55,96,234,234,162,0,142,96,3,96,144
194 DATA 20,61,201,96,208,3,108,231,0,201,69,208,3,76,96,60,105
195 DATA 201,70,240,249,201,71,240,245,201,72,240,241,32,130,60,96,29
196 DATA 173,1,255,41,3,174,96,3,157,96,55,32,157,60,96,0,119
197 DATA 206,96,3,76,130,60,169,2,141,79,3,96,0,0,0,206,243
198 DATA 96,3,162,0,142,97,3,96,174,97,3,224,1,240,3,76,137
199 DATA 218,59,32,160,60,169,0,141,97,3,76,194,60,76,130,60,255
200 DATA 173,78,3,174,96,3,157,144,63,96,160,0,140,97,3,140,247
201 DATA 96,3,76,206,60,32,208,62,172,96,3,185,96,55,141,98,53
202 DATA 3,96,172,96,3,173,98,3,153,96,55,76,109,59,140,99,151
203 DATA 3,76,109,59,174,99,3,224,1,240,11,162,1,142,99,3,126
204 DATA 174,96,3,76,99,60,96,160,0,140,99,3,76,33,59,162,56
205 DATA 248,162,254,160,60,142,76,3,140,77,3,76,0,56,32,139,92
206 DATA 57,162,13,160,61,134,212,132,213,76,124,58,234,32,216,56,148
207 DATA 32,14,57,96,234,234,234,162,255,232,189,112,55,157,52,3,70
208 DATA 189,116,55,157,72,3,189,120,55,157,56,3,189,124,55,157,161
209 DATA 64,3,224,3,208,227,173,128,55,133,210,173,129,55,133,211,81
210 DATA 162,0,32,241,60,162,255,232,189,52,3,157,112,55,189,56,165
211 DATA 3,157,120,55,189,64,3,157,124,55,189,72,3,157,116,55,239
212 DATA 224,3,208,227,165,210,141,128,55,165,211,141,129,55,96,162,16
213 DATA 1,142,85,3,96,32,216,56,162,223,160,14,142,128,55,134,113
214 DATA 210,140,129,55,132,211,234,234,234,96,201,74,240,3,76,223,188
215 DATA 58,108,231,0,169,7,141,11,54,169,160,141,19,54,169,6,217
216 DATA 141,30,54,169,30,141,31,54,169,120,141,38,54,169,0,141,202
217 DATA 39,54,169,55,141,43,54,169,51,141,44,54,169,63,141,45,152
218 DATA 54,169,224,141,51,54,169,192,141,52,54,169,255,141,53,54,181
219 DATA 169,30,141,62,54,169,0,141,63,54,169,96,141,70,54,169,46
220 DATA 120,141,71,54,169,54,141,27,54,169,224,141,35,54,234,234,130
221 DATA 234,234,234,234,234,234,169,126,141,83,54,141,84,54,169,129,250
222 DATA 141,112,54,169,66,141,113,54,169,66,141,118,54,169,129,141,45
223 DATA 119,54,96,169,5,141,11,54,169,224,141,19,54,169,30,141,60
224 DATA 30,54,169,0,141,31,54,169,96,141,38,54,169,120,141,39,166
225 DATA 54,169,7,141,43,54,169,3,141,44,54,169,255,141,45,54,7
226 DATA 169,236,141,51,54,169,204,141,52,54,169,252,141,53,54,169,61
227 DATA 6,141,62,54,169,30,141,63,54,169,120,141,70,54,169,0,163
228 DATA 141,71,54,169,6,141,27,54,169,236,141,35,54,234,234,234,208
229 DATA 234,234,234,234,234,169,102,141,83,54,141,84,54,169,0,141,4
230 DATA 112,54,169,0,141,113,54,169,0,141,118,54,169,0,141,119,18
231 DATA 54,96,165,217,201,0,208,1,96,32,216,56,198,217,96,162,223
232 DATA 0,134,218,230,218,165,218,197,219,240,6,32,170,60,76,163,42
233 DATA 62,32,20,61,32,232,62,96,169,0,234,234,234,141,80,3,156
234 DATA 141,81,3,141,82,3,141,83,3,141,85,3,141,84,3,96,207
235 DATA 202,202,202,202,202,202,202,202,202,202,202,96,232,232,232,232,174
236 DATA 232,232,232,232,232,232,232,96,166,228,232,138,41,15,170,224,118
237 DATA 14,208,3,162,0,138,188,130,55,140,200,15,133,228,96,31,205
238 A=(I-14336)/16:PRINT"?DATA ERROR IN";A+126;
239 END
240 PRINT"DAS PROGRAMM ZERSTOERT SICH VON SELBST"
241 PRINT"BEI FEHLERFREIEM DURCHLAUF. TIPPEN SIE"
242 PRINT"PROGRAMMTEIL 2 EIN. WENN SIE DIESES"
243 PRINT"PROGRAMM NACH DEM ZERSTOEREN AB'SAVE'N"
244 PRINT"WOLLEN, GEBEN SIE MONITOR EIN. DANACH"
245 PRINT"TIPPEN SIE S "+CHR$(34)+"ZS,MC"+CHR$(34)+",1,3400,3F00 EIN."
246 PRINT"ZUM LADEN MONITOR L EINGEBEN"
247 NEW
```

Horror-Castle – Teil Zwei

```
118 RESTORE:FORI=13312TO14103STEP12:SU=0
119 FORJ=0TO11
120 READZ:SU=SU+ZAND255
121 POKEI+J,Z
122 NEXTJ
123 READS:IFS<>SUTHEN126
124 NEXTI
125 GOTO128
126 A=(I-13312)/12:PRINT"?DATA ERROR IN";A+135;
127 END
128 PRINT"■DAS PROGRAMM ZERSTOERT SICH VON SELBST■"
129 PRINT"BEI FEHLERFREIEM DURCHLAUF. TIPPEN SIE■"
130 PRINT"PROGRAMMTEIL 3 EIN. WENN SIE DIESES■"
131 PRINT"PROGRAMM NACH DEM ZERSTOEREN AB'SAVE'N■"
132 PRINT"WOLLEN, GEBEN SIE ■MONITOR■ EIN. DANACH■"
133 PRINT"TIPPEN SIE ■S "+CHR$(34)+"ZS,MC"+CHR$(34)+",1,3400,3F00■ EIN.■"
134 PRINT"ZUM LADEN ■MONITOR■ ■L■ EINGEBEN■■■":NEW
135 DATA 0,0,0,0,0,0,0,0,0,126,102,228:REM "@/A"
136 DATA 126,102,102,0,0,0,126,102,126,102,126,0,144:REM "A/B"
137 DATA 0,0,126,96,96,96,126,0,0,0,124,102,254:REM "C/D"
138 DATA 102,102,126,0,0,0,126,96,120,96,126,0,126:REM "D/E"
139 DATA 0,0,126,96,120,96,96,0,0,0,126,96,244:REM "F/G"
140 DATA 110,102,126,0,0,0,102,102,126,102,102,0,104:REM "G/H"
141 DATA 0,0,24,24,24,24,24,0,0,0,24,24,168:REM "I/J"
142 DATA 24,24,120,0,0,0,102,108,120,108,102,0,196:REM "J/K"
143 DATA 0,0,96,96,96,96,126,0,0,0,102,126,226:REM "L/M"
144 DATA 102,102,102,0,0,0,102,118,110,102,102,0,72:REM "M/N"
145 DATA 0,0,126,102,102,102,126,0,0,0,126,102,18:REM "O/P"
146 DATA 126,96,96,0,0,0,126,102,102,110,127,0,117:REM "P/Q"
147 DATA 0,0,126,102,126,108,102,0,0,0,126,96,18:REM "R/S"
148 DATA 126,6,126,0,0,0,126,24,24,24,24,0,224:REM "S/T"
149 DATA 0,0,102,102,102,102,126,0,0,0,102,102,226:REM "U/V"
150 DATA 102,60,24,0,0,0,102,102,102,126,102,0,208:REM "V/W"
151 DATA 0,0,102,60,24,60,102,0,0,0,102,102,40:REM "X/Y"
152 DATA 62,6,126,0,0,0,126,12,24,48,126,0,18:REM "Y/Z"
153 DATA 0,0,60,48,48,48,60,0,40,68,154,178,192:REM "[/\"
154 DATA 154,68,40,0,0,0,60,12,12,12,60,0,162:REM "\/]"
155 DATA 0,0,0,0,0,0,0,0,0,0,0:REM "^/_"
156 DATA 0,0,0,0,0,0,0,0,0,0,0:REM "_/¯"
157 DATA 0,0,24,24,24,0,24,0,0,0,24,24,144:REM "!/""
158 DATA 0,0,0,0,102,0,126,102,126,102,102,0,148:REM ""/#"
159 DATA 102,0,126,102,102,102,126,0,102,0,102,102,198:REM "$/%"
160 DATA 102,102,126,0,0,0,60,102,60,102,62,0,204:REM "%/&"
161 DATA 0,0,0,0,0,0,0,0,0,12,24,36:REM "'/("
162 DATA 24,24,12,0,0,0,48,24,24,24,48,0,228:REM "(/)"
163 DATA 0,102,60,255,60,102,0,0,0,0,0,67:REM "*/+"
164 DATA 0,0,0,0,0,0,0,0,0,24,48,72:REM "+/,"
165 DATA 0,255,0,0,0,0,0,0,0,0,255:REM "-/."
166 DATA 0,0,24,0,0,0,6,12,24,48,96,0,210:REM ".//"
167 DATA 0,0,126,102,102,102,126,0,0,0,56,24,126:REM "0/1"
168 DATA 24,24,24,0,0,0,126,6,126,96,126,0,40:REM "1/2"
169 DATA 0,0,126,6,30,6,126,0,0,0,96,102,236:REM "3/4"
170 DATA 126,6,6,0,0,0,126,96,126,6,126,0,106:REM "4/5"
171 DATA 0,0,126,96,126,102,126,0,0,0,126,6,196:REM "6/7"
172 DATA 30,6,6,0,0,0,126,102,126,102,126,0,112:REM "7/8"
173 DATA 0,0,126,102,126,6,126,0,0,0,0,24,254:REM "9/:"
174 DATA 0,24,0,0,0,0,0,0,0,0,0,24:REM ":;"
175 DATA 0,0,30,62,126,62,30,0,0,0,0,0,54:REM "</="
```

```
176 DATA 0,0,0,0,0,0,120,124,126,124,120,0,102:REM "=/>"
177 DATA 0,0,60,12,24,0,24,0,0,0,124,96,84:REM "?/—"
178 DATA 102,96,124,0,7,3,3,7,15,7,1,15,124:REM "—/♠"
179 DATA 0,128,192,160,240,224,128,240,27,51,51,54,215:REM "|/—"
180 DATA 6,6,6,30,216,204,204,224,96,96,120,0,184:REM "—/—"
181 DATA 3,7,5,55,51,63,7,15,192,224,160,224,238:REM "‾/_"
182 DATA 192,255,224,240,31,31,31,15,6,6,30,0,37:REM "_/|"
183 DATA 248,248,248,240,96,96,96,120,0,0,0,24,136:REM "|/╮"
184 DATA 24,0,0,0,0,126,126,126,126,126,126,0,12:REM "╮/╰"
185 DATA 0,0,0,0,0,0,0,0,126,126,126,122:REM "╯/└"
186 DATA 126,126,126,0,0,0,0,0,0,0,0,0,122:REM "└/╲"
187 DATA 129,66,24,60,60,24,66,129,12,24,60,126,12:REM "╱/┌"
188 DATA 126,126,60,0,126,96,96,96,96,96,126,0,20:REM "┌/┐"
189 DATA 126,102,102,126,102,102,102,0,126,96,96,126,182:REM "●/_"
190 DATA 6,6,126,0,126,24,24,24,24,24,24,0,152:REM "_/♥"
191 DATA 96,96,96,96,96,96,126,0,126,96,96,126,122:REM "|/╭"
192 DATA 96,96,126,0,102,102,102,126,102,102,102,0,32:REM "╭/╳"
193 DATA 126,102,102,102,102,102,126,0,126,102,102,126,194:REM "○/♣"
194 DATA 108,108,102,0,126,102,102,126,108,108,102,0,68:REM "♣/|"
195 DATA 126,102,102,102,102,102,126,0,255,0,255,0,248:REM "♦/+"
196 DATA 255,0,255,0,0,255,0,255,0,255,0,255,250:REM "+/▒"
197 DATA 255,0,63,0,15,0,3,0,255,0,252,0,75:REM "|/π"
198 DATA 240,0,192,0,1,0,7,0,31,0,127,0,86:REM "π/◥"
199 DATA 0,0,0,0,0,0,0,0,128,0,224,0,96:REM " /▌"
200 DATA 248,0,254,0,0,0,0,0,0,0,0,0,246:REM "▌/▄"
```

★ Horror-Castle – Teil Drei ★ – ★ Horror-Castle – Teil Drei ★

```
120 IFPEEK(16127)=31THEN122
121 CLR:POKE55,0:POKE56,52:CLR:GOSUB376
122 POKE65299,PEEK(65299)AND3ORS2:VOL8
123 POKE65298,PEEK(65298)AND251:COLOR4,1:COLOR0,1:DIMHS$(6),HS(6),LE(6)
124 FORI=1TO5:HS$(I)="R&R":HS(I)=0:LE(I)=1:NEXT:GOSUB221:GOSUB126
125 P=0:L=5:GOTO227
126 PRINT"                         ◥+▌
127 PRINT"▒                        ◥π |▌
128 PRINT"                       +[R&R▌+
129 PRINT"              ++++++++++   ++++++++++
130 PRINT"              +        |▌ ◥π        +
131 PRINT"              + ▌/// ///  ▌|+π ▌/// ///  ▌+
132 PRINT"              + ▌/    / /  ▌+  ▌/     /  ▌+
133 PRINT"              + ▌/// / /   ▌+  ▌//    /  ▌+
134 PRINT"              + ▌  / / /   ▌+  ▌/     /  ▌+
135 PRINT"              + ▌/// ///   ▌+  ▌/     /  ▌+
136 PRINT"              +          +          +
137 PRINT"              +++++++▌PRESENTS:▌+++++++
138 PRINT"              +                      ▌+
139 PRINT"              +  ▌┌●_♥ ╭ OF ╳○♣ |♦|  ▌ +
140 PRINT"              +  ▌├───────────────    ▌+
141 PRINT"              +                      +
142 PRINT"               |▌    ▌BY ▌CHR ▌RASKOB   ▌◥π
143 PRINT"                |▌                 ◥π
144 PRINT"                 |▌               ◥π
145 PRINT"                  |▌     ▌|||   ▌  ◥π
146 PRINT"                   |▌    ▌|||   ▌◥π
147 PRINT"                    |▌          ◥π
148 PRINT"                     |++++++++++π
```

Schon gemerkt? Noch nicht? Wir verraten es Ihnen!

Ihr Heft steht auf dem Kopf (ähem).

```
149 PRINT"
150 CHAR1,11,24,"[illegible] 10/1985 BY R & R[illegible]":GOSUB326:RETURN
151 PRINT"[illegible]":SYS15764
152 PRINT"[illegible]
153 PRINT"[illegible]
154 PRINT"[illegible]
155 PRINT"[illegible]
156 PRINT"[illegible]
157 PRINT"[illegible]
158 PRINT"[illegible]
159 PRINT"[illegible]
160 PRINT"[illegible]
161 PRINT"[illegible]
162 PRINT"[illegible]
163 PRINT"[illegible]
164 PRINT"[illegible]
165 PRINT"[illegible]
166 PRINT"[illegible]
167 PRINT"[illegible]
168 PRINT"[illegible]
169 PRINT"[illegible]
170 PRINT"[illegible]
171 PRINT"[illegible]
172 PRINT"[illegible]
173 PRINT"[illegible]
174 PRINT"[illegible]
175 PRINT"[illegible]"
176 RETURN
177 DATA0,0,0,0,0,0,0,0
178 DATA16,2,0,3,0,3,0,3
179 DATA0,0,0,0,0,0,0,0
180 DATA-1
181 DATA0,0,0,0,0,0,0,0
182 DATA0,0,0,0,0,0,0,0
183 DATA0,0,0,0,0,0,0,0
184 DATA0,0,0,0,0,0,0,0
185 DATA0,0,0,0,0,0,0,0
186 DATA0,0,0,0,0,0,0,0
187 DATA-1
188 DATA69,69,69,69,69,69,69,69,69,69,69,69,69,69,69,69
189 DATA70,70,70,70,70,70,70,70,70,70,70,70,70,70,70,70
190 DATA71,71,71,71,71,71,71,71,71,71,71,71,71,71,71,71
191 DATA72,72,72,72,72,72,72,72,72,72,72,72,72,72,72,72
192 DATA119,103,87,71,117,101,85,69,118,102,86,70,116,100,84,68
193 DATA119,103,87,71,117,101,85,69,118,102,86,70,116,100,84,68
194 DATA119,103,87,71,117,101,85,69,118,102,86,70,116,100,84,68
195 DATA119,103,87,71,117,101,85,69,118,102,86,70,116,100,84,68
196 DATA73,73,73,73,73,73,73,73,73,73,73,73,73,73,73,73
197 DATA73,73,73,73,73,73,73,73,73,73,73,73,73,73,73,73
198 DATA73,73,73,73,73,73,73,73,73,73,73,73,73,73,73,73
199 DATA73,73,73,73,73,73,73,73,73,73,73,73,73,73,73,73
200 DATA99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99
201 DATA99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99
202 DATA99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99
203 DATA99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,-1
204 DATA39,39,39,39,39,39,39,39,39,39,39,39,39,39,39,39
205 DATA13,13,13,13,13,13,13,13,13,13,13,13,13,13,13,13
206 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Na, Leute, ist nicht allein das Abtippen schon ganz schön schaurig?

```
207 DATA65,66,67,68,121,121,121,121
208 DATA32,32,32,32,32,32,32,32
209 DATA223,14,5,24,20,18,1,32,19,16,5,3,9,1,12,32,32,32,32
210 DATA-1
211 RESTORE168:A=208
212 READZ:IFZ<>-1THENPOKEA,Z:ELSE214
213 A=A+1:GOTO212
214 A=819
215 READZ:IFZ<>-1THENPOKEA,Z:ELSE217
216 A=A+1:GOTO215
217 A=16128:RESTORE188
218 READZ:IFZ<>-1THENPOKEA,Z:ELSE220
219 A=A+1:GOTO218
220 GOTO384
221 POKE65286,PEEK(65286)AND239
222 RETURN
223 POKE65286,PEEK(65286)OR16
224 RETURN
225 POKE239,0:WAIT239,1:POKE239,0
226 RETURN
227 :
228 GOSUB221:GOSUB211:GOSUB311:
    GOSUB221:GOSUB151
229 GOSUB223:GOSUB225
230 E1$="EXTRA SPECIAL    "
231 E2$="                 "
232 H=PEEK(217)
```

```
233 EX=1:SP%(0)=15891:SP%(1)=15764
234 M=1:ML=1:SY=16031
235 CHAR1,33,6,"▣VAMP.":POKE217,PEEK(217)+3AND255
236 CHAR1,33,9,"HENRYS"
237 CHAR1,33,12,"HELPS":CHAR1,33,15,"HIGH":CHAR1,32,16,LEFT$(STR$(HS(1)),7)+"▣"
238 CHAR1,32,10,STR$(L)+" ":IFFL=-1THENM=10:ML=10
239 CHAR1,32,7,STR$(PEEK(216))+" "
240 CHAR1,32,13,STR$(H)+" "
241 CHAR1,9,0,"SCORE"+STR$(P):CHAR1,33,21,"▣▣▣▣▣":CHAR1,33,22,"OF":CHAR1,33,23,
"▣▣▣▣▣▣"
242 CHAR1,8,24,"▣E▣ "+E2$:FL=0:CHAR1,33,18,"\ 1985":CHAR1,33,19,"BY R&R":G=0
243 GOTO247
244 CHAR1,10,12,"▣▣ GAME  OVER ▣":FORI=1TO1000:NEXTI
245 CHAR1,32,10," O "
246 GOSUB225:GOTO273
247 SYS16056
248 SYSSY:IFPEEK(853)<>0ORPEEK(847)=1THENSYS16056
249 P=P+PEEK(848)*M:G=G+PEEK(848):IFPEEK(848)>0THENSOUND3,300,1:SOUND3,0,0
250 IFPEEK(849)>0THENP=P+PEEK(849)*M*50:G=G+PEEK(849):SOUND1,900,5:GOSUB388
251 IFPEEK(850)>0THENP=P+PEEK(850)*M*100:G=G+PEEK(850):SOUND1,200,5:GOSUB258
252 CHAR1,32,13,STR$(PEEK(217))+" "
253 IFPEEK(847)<>0THEN266
254 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣▣▣"P
255 IFG>=454THEN269
256 S=S+1AND1:SYSSP%(S)
257 GOTO247
258 EX=PEEK(228)+1:MID$(E2$,INT(EX+.5),1)=MID$(E1$,INT(EX+.5),1):CHAR1,10,24,E2$
259 IFLEFT$(E2$,5)="EXTRA"THENL=L+1:CHAR1,32,10,STR$(L)+" ":GOTO262
260 IFMID$(E2$,7,7)="SPECIAL"THEN264
261 RETURN
262 MID$(E2$,1,5)="     ":G=500
263 GOTO260
264 MID$(E2$,7,7)="       ":P=P+INT(RND(1)*6+15)*1000
265 GOTO309
266 L=L-1:IFL<>0THENPOKE847,0:FORI=1TO1000:NEXT:POKE217,PEEK(217)+1AND255:ELSE24
4
267 SYS16018:CHAR1,32,10,STR$(L)+" ":POKE239,0:WAIT239,1:POKE239,0
268 GOTO254
269 IFPEEK(216)<16THENPOKE216,PEEK(216)+1
270 GOSUB221:GOSUB151:GOSUB217
271 GOSUB223:GOSUB225
272 GOTO232
```

```
273 FORI=1TO5:IFP>HS(I)THENGOSUB300:GOTO275
274 NEXT:GOTO293
275 FORJ=5TOISTEP-1:HS(J+1)=HS(J)
276 HS$(J+1)=HS$(J):LE(J+1)=LE(J)
277 NEXTJ:A$="":FORD=1TO3:CHAR1,10+D,9,"A":B=65
278 A=JOY(1)
279 IFA=1THENA=1:GOTO283
280 IFA=5THENA=-1:GOTO283
281 IFA=128THEN290
282 GOTO278
283 B=B+A:IFB=64THENB=32
284 IFB=34THENB=90
285 IFB=31THENB=37
286 IFB=33THENB=65
287 IFB=38THENB=32:ELSEIFB=91THENB=35
288 CHAR1,10+D,9,CHR$(B):FORJ=1TO100:NEXT
289 GOTO278
290 A$=A$+CHR$(B):B=65:IFD<>3THENCHAR1,10+D+1,9,"A":FORJ=1TO50:NEXT
291 NEXTD
292 HS$(I)=A$:HS(I)=P:LE(I)=PEEK(216)+PEEK(219)-3:GOTO293
293 GOSUB306
294 CHAR1,10,6,"  INI  SCORE  LE"
295 FORI=1TO5
296 CHAR1,9,I+8,STR$(I)+" "+HS$(I):CHAR1,15,I+8,STR$(HS(I))
297 CHAR1,22,I+8,STR$(LE(I))
298 NEXT:CHAR1,10,15,"YOUR SCORE :"+STR$(P):GOSUB223:GOSUB326
299 GOTO125
300 PRINT"▒▒▒"
301 CHAR1,11,3,"GIVE  IN  YOUR"
302 CHAR1,11,4,"   INITIALS   "
303 CHAR1,11,6,"CONGRATULATION"
304 CHAR1,15,9,STR$(P)
305 RETURN
306 PRINT"▒▒▒":GOSUB221
307 CHAR1,10,3,"▒▒▒ TOPFIVE ▒▒▒"
308 RETURN
309 FL=-1:G=500
310 RETURN
311 :
312 PRINT"▒"
313 PRINT"▒▒▒":GOSUB221
314 B=1:CHAR1,3,1,"NUMBER OF VAMPIRES"
315 CHAR1,5,6,"1▒▒2▒▒3▒▒4▒▒5▒▒6"
316 CHAR1,3,6,"> ▒ <":GOSUB223
317 A=JOY(1):IFPEEK(1347)=7ANDPEEK(2038)=0THEN389
318 IFA=1THENA=-1:GOTO322
319 IFA=5THENA=1:GOTO322
320 IFA=128THENPOKE216,B:FORI=1TO100:NEXT:RETURN
321 GOTO317
322 B=B+A:IFB=0THENB=1:GOTO317
323 IFB=7THENB=6:GOTO317
324 FORI=1TO8:CHAR1,3,5+I,"  ▒  ":NEXT
325 CHAR1,3,5+B,"> ▒ <":GOTO317
326 RESTORE331:GOSUB223:POKE239,0
327 READX,Y,Z:IFX=-1THENSOUND2,0,1:RESTORE331:RETURN
328 SOUNDX,0,1:SOUNDX,Y,Z*5:IFX=1THENSYS15891:ELSESYS15764
329 IFPEEK(239)<>0ANDX=1THENSOUND1,0,0:SOUND2,0,0:RETURN
330 GOTO327
331 DATA1,770,2,2,810,2
332 DATA1,770,2,2,810,2
333 DATA1,770,2,2,810,2
334 DATA1,770,4,2,810,4
335 DATA1,785,2,2,822,2
336 DATA1,770,4,2,810,4
337 DATA1,739,2,2,785,2
338 DATA1,704,2,2,770,2
339 DATA1,739,2,2,785,2
340 DATA1,739,2,2,785,2
341 DATA1,739,2,2,785,2
342 DATA1,739,4,2,785,4
343 DATA1,770,2,2,810,2
344 DATA1,739,4,2,785,4
345 DATA1,704,2,2,770,2
346 DATA1,685,2,2,739,2
347 DATA1,704,2,2,770,2
348 DATA1,704,2,2,770,2
349 DATA1,704,2,2,770,2
350 DATA1,704,4,2,770,4
351 DATA1,739,2,2,785,2
352 DATA1,704,4,2,770,4
353 DATA1,685,2,2,739,2
354 DATA1,643,2,2,704,2
355 DATA1,685,4,2,739,4
356 DATA1,685,4,2,810,4
357 DATA1,685,8,2,854,8
358 DATA1,798,2,2,834,2
359 DATA1,798,2,2,834,2
360 DATA1,798,2,2,834,2
361 DATA1,798,4,2,834,4
362 DATA1,810,2,2,854,2
363 DATA1,798,2,2,834,2
364 DATA1,770,2,2,810,2
365 DATA1,739,4,2,798,4
366 DATA1,704,2,2,770,2
367 DATA1,704,2,2,770,2
368 DATA1,704,2,2,770,2
369 DATA1,704,4,2,770,4
370 DATA1,739,2,2,798,2
371 DATA1,704,2,2,770,2
372 DATA1,685,2,2,739,2
373 DATA1,643,4,2,704,4
```

```
374 DATA1,596,8,2,685,8
375 DATA-1,-1,-1,-1,0,0
376 RESTORE380
377 I=16288
378 READA$:IFA$="$"THENSYS16288:RETURN
379 POKEI,DEC(A$):I=I+1:GOTO378
380 DATAA9,01,A2,01,A0,01,20,BA,FF
381 DATAA9,05,A2,BC,A0,3F,20,BD,FF
382 DATAA9,00,A2,00,A0,34,20,D5,FF
383 DATA60,5A,53,2C,4D,43,$
384 A=14144
385 READZ:IFZ<>-1THENPOKEA,Z:ELSE387
386 A=A+1:GOTO385
387 RETURN
388 ML=ML+.5:M=INT(ML):RETURN
389 PRINT"□"
390 CHAR1,12,3,"CUBIK, ANYONE ?"
391 DOUNTILJOY(1)=128:LOOP
392 CHAR1,12,7,"-"
393 GETA$:A=ASC(A$)-48:IFA<0ORA>1THEN393
394 CHAR1,12,6,A$:ONA+1GOTO395,398
395 CHAR1,12,7," -"
396 GETA$:A=ASC(A$)-48:IFA<1ORA>9THEN396:ELSECHAR1,13,6,A$
397 POKE216,A:GOTO221
398 CHAR1,12,7," -"
399 GETA$:A=ASC(A$)-48:IFA<0ORA>6THEN399:ELSECHAR1,13,6,A$
400 POKE216,A+10:GOTO221
```

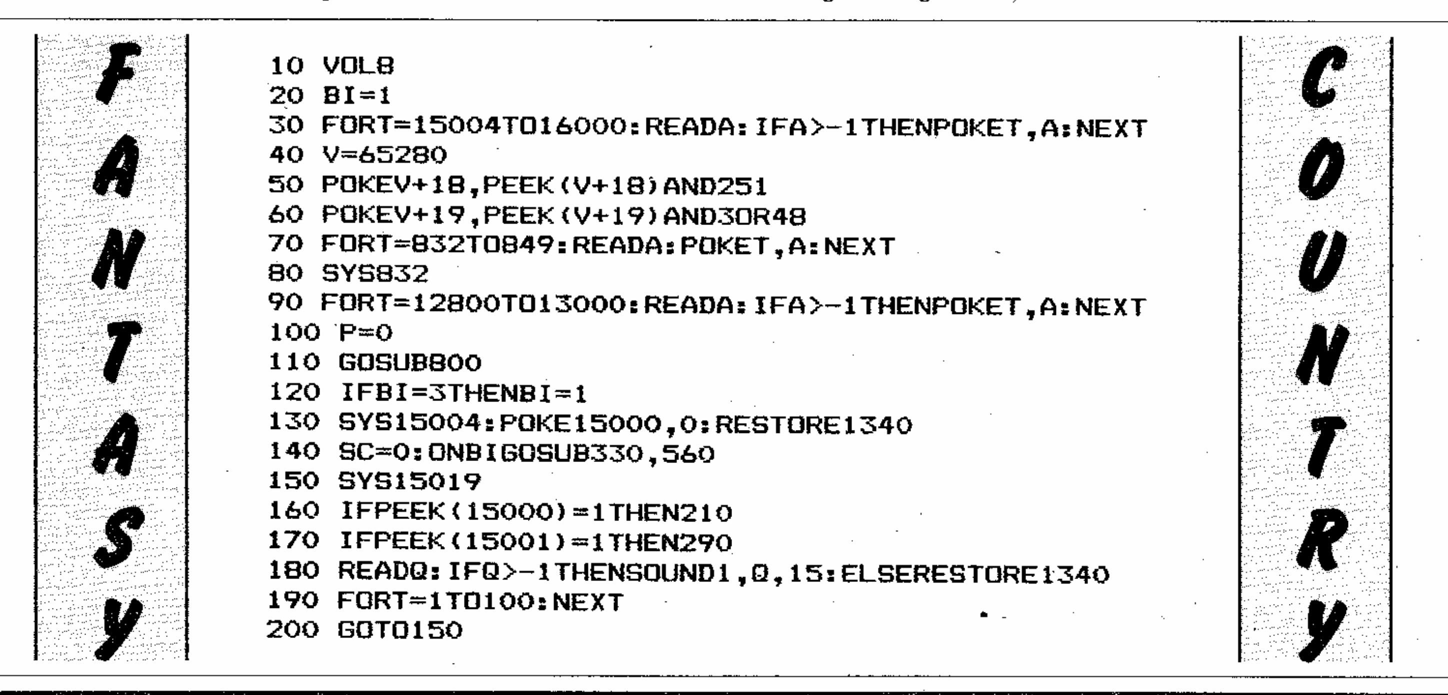
Mit dem C-16-Computer auf ins „Fantasy-Country"

Hier sind wieder kurze Reaktionszeiten und ausgeprägte Beweglichkeit gefordert! Auf Ihrem Bildschirm „wimmelt" es von gewandten Feinden und heimtückischen Hindernissen. (S. G.)

Sie sind mit Ihrem Raumschiff im „Fantasy-Country" gelandet. Um sich hier am Leben zu erhalten, muß man die geschickt versteckten Sterne, die im ganzen Land verteilt sind, einsammeln. Die Suche nach den begehrten Gegenständen ist – wie nicht anders zu erwarten – mit manchen Gefahren verbunden. Auf dem gefahrvollen Weg, der zum Ziel führt, muß man, sich nach rechts und links bewegenden Pfeilen und sich nach oben und unten bewegenden Schattenköpfen, ausweichen. Schon die erste Berührung mit einem dieser „Objekte" setzt Ihrem Leben ein Ende.
Um die High-Score-Anzeige auszuführen, müssen so viele Sterne wie möglich eingesammelt werden.

Die Steuerung erfolgt durch die Cursortasten:

<A> = nach oben
<Z> = nach unten
<.> = nach rechts
<,> = nach links




```
10 VOL8
20 BI=1
30 FORT=15004TO16000:READA:IFA>-1THENPOKET,A:NEXT
40 V=65280
50 POKEV+18,PEEK(V+18)AND251
60 POKEV+19,PEEK(V+19)AND3OR48
70 FORT=832TO849:READA:POKET,A:NEXT
80 SYS832
90 FORT=12800TO13000:READA:IFA>-1THENPOKET,A:NEXT
100 P=0
110 GOSUB800
120 IFBI=3THENBI=1
130 SYS15004:POKE15000,0:RESTORE1340
140 SC=0:ONBIGOSUB330,560
150 SYS15019
160 IFPEEK(15000)=1THEN210
170 IFPEEK(15001)=1THEN290
180 READQ:IFQ>-1THENSOUND1,Q,15:ELSERESTORE1340
190 FORT=1TO100:NEXT
200 GOTO150
```

Dieses Programm stellt selbst für die erfahrensten Spiele-Freaks eine Herausforderung dar!

```
210 RESTORE1400
220 READA,B:IFA>-1THENSOUND1,A,B:GOTO220:ELSEGOTO230
230 COLOR0,1,1:PRINT"G A M E   O V E R"
240 IFP>HITHENHI=P:PRINT"NEUER HISCORE : ";HI:GOTO260
250 PRINT"PUNKTE ";P;"  HISCORE : ";HI
260 POKE239,0
270 GETA$:IFA$=""THEN270
280 GOTO100
290 P=P+1:PRINT"PUNKTE :";P;"  HISCORE : ";HI;""
300 POKE15001,0:SC=SC+1:IFSC=10THENBI=BI+1:GOTO310:ELSE150
310 SOUND1,169,10:SOUND1,596,10:SOUND1,810,10:SOUND1,169,10
320 GOTO120
330 COLOR0,14,6:COLOR4,1,1
340 PRINT"                                   "
350 PRINT"                                   "
360 PRINT"  "SPC(29)"   ":PRINT"    "SPC(15)" "SPC(15)"  "
370 PRINT" "SPC(38)"  ":PRINT"
380 PRINT" "SPC(38)"  ":PRINT"   "SPC(18)"  "SPC(14)"  "
390 PRINT" "SPC(22)" "SPC(15)"  ":PRINT" "SPC(11)"|        "SPC(13
)"  "
400 PRINT"                    "SPC(13)"  "
410 PRINT" "SPC(11)"  |      "SPC(13)"  "
420 PRINT"  -|               "SPC(13)"  "
430 PRINT"                                 "
440 PRINT"                                 "
450 PRINT"                                 "
460 PRINT" "SPC(38)"  ":PRINT"        "SPC(26)"|    "
470 PRINT"       "SPC(28)"     "
480 PRINT"       "SPC(12)"   "SPC(16)"    "
490 PRINT"  Z          "
500 PRINT"  I          |||||                  "
510 PRINT"  E          |||                    "
520 PRINT"  L                                 "
530 PRINT"                                    "
540 PRINT"                                  ";
550 RETURN
560 COLOR0,6,5:COLOR4,1,1
570 PRINT"                                  "
580 PRINT"                                  "
590 PRINT" "SPC(38)"  ":PRINT"       -|"SPC(13)"  "SPC(10)"    "
600 PRINT" "SPC(22)"       /    ":PRINT" "SPC(22)"
 "
610 PRINT"     |  "SPC(13)"        "
620 PRINT"         |||       "SPC(9)"_    "
630 PRINT"         |||||     "SPC(13)"  "
640 PRINT"         |||       "SPC(13)"  "
650 PRINT"              L    "SPC(13)"  "
660 PRINT"                                 "
670 PRINT"                                 "
680 PRINT"  _"SPC(11)"_                    "
690 PRINT"  _"SPC(18)"_ _"SPC(10)"   "
700 PRINT"     "SPC(17)"_"SPC(11)"     "
710 PRINT"     "SPC(27)"    ":PRINT"      "SPC(28)"    "
720 PRINT"        "SPC(11)"            "
730 PRINT"                                 "
740 PRINT"     Z    |||||                  "
750 PRINT"  |  I    |||                    "
760 PRINT"     E    L                      "
770 PRINT"       L                         "
```




```
780 PRINT"                                     ";
790 RETURN
800 COLOR0,6,5:COLOR4,2,7
810 PRINT"T I G E R   -   S O F T"
820 PRINT"PRAESENTIERT....."
830 PRINT"P H A N T A S I E  -  LAND"
840 PRINT"DU STEHST MIT DEINEM RAUMSCHIFF IN"
850 PRINT"EINEM PHANTASIELAND MIT VIELEN FI-"
860 PRINT"GUREN . WEICHE DEN PFEILEN - UND"
870 PRINT"DEN SCHATTENKOEPFEN - AUS UND HOLE"
880 PRINT"SOVIEL ALS MOEGLICH STERNE  ."
890 PRINT"BEVOR DU GETROFFEN WIRST , ODER AN"
900 PRINT"EIN HOCHHAUS KNALLST SOLLTEST DU"
910 PRINT"SOVIEL WIE MOEGLICH STERNE BESITZEN."
920 PRINT"GESTEUERT WIRD ; <A> : RAUF, <Z>"
930 PRINT" : RUNTER , <.> : RECHTS UND <,> :"
940 PRINT"LINKS . VIEL SPASS"
950 POKE239,0
960 GETA$:IFA$=""THEN960
970 RETURN
980 REM ===MASCHINENCODEDATAS =====
990 DATA169,25,141,154,58,133,253,169,13,141,155,58,133,254,96,160,0,169,32,145,
253
1000 DATA200,145,253,160,40,145
1010 DATA253,200,145,253,169,12,133,254,169,0,133,253,160,40,177,253,201,71,240,
28,201
1020 DATA69,240,50
1030 DATA201,68,240,61,230,253,208,2,230,254,165,254,201,15,208,228,165,253,201,
192
1040 DATA208,222,240,59,169,32,145,253,160,0,177,253,201,32,208,6,169,71,145,253
,208
1050 DATA218,160,40,169,70,145,253,208,210,173,0,255,201,16,176,203,160,39,169,6
8
1060 DATA145,253,144,195,169,32,145,253,160,39,177,253,201,32,208,183,169,68,145
,253
1070 DATA208,177,169,15,133,254,169,231,133,253,160,0,177,253,201,70,240,26,201,
69
1080 DATA240,48,201,73,240,59,198,253,165,253,201,255,208,232,198,254,165,254,20
1,11
1090 DATA208,224,240,59,169,32,145,253,160,40,169,70,145,253,160,80,177,253,201
1100 DATA32,240,220,160,40,169,71,145,253,208,212,173,0,255,201,16,176,205,160,1
1110 DATA169,73,145,253,144,197,169,32,145,253,160,1,177,253,201,32,208,185,169
1120 DATA73,145,253,208,179,173,154,58,133,253,173,155,58,133,254,165,198,201,10
,240,14
1130 DATA201,12,240,21,201,47,240,28
1140 DATA201,44,240,34,208,121,165,253,56,233,40,133,253,144,29,176,33,165,253
1150 DATA24,105,40,133,253,176,22,144,22,198,253,165,253,201,255,240,8,208,12,23
0,253
1160 DATA240,6,208
1170 DATA6,198,254,208,2,230,254,165,253,141,154,58,165,254,141,155,58,160,0,32
1180 DATA240,59,200,32,240,59,160,40,32,240,59,200,32,240,59,76,7,60,177,253,201
,32
1190 DATA240,16,234,201,74,208,6,169,1,141,153,58,96,169,1,141,152,58,96,160,0,1
69
1200 DATA64,145,253,200,169,65,145,253,160,40,169,66,145,253,200,169,67,145,253
1210 DATA96,169,0,162,0,24,105,1,208,251,232,224,17,208,246,240,173,-1
1220 REM =====ZEICHENSATZDATAS=====
1230 DATA162,0,189,0,208,157,0,48,189,0,209,157,0,49,202,208,241,96
```

An alle C-16-Programmautoren!

Bei unserer Leserschaft herrscht eine riesige Nachfrage nach C-16-Programmen. – Also, auf ans Werk und „bombardiert“ die Redaktion mit euren Programmbeiträgen.

Wir zahlen <u>120,- DM</u> pro komplett abgedruckte Seite! Alles weitere auf S. 12.

```
1240 DATA3,12,48,255,255,255,255,63,0,192,48,252,252,252,252,240
1250 DATA31,19,16,16,31,4,68,124,224,32,32,32,224,128,136,248
1260 DATA0,48,96,255,255,255,96,48,60,66,126,255,126,66,60,0,60,126,255,255,126,
60
1270 DATA255,255,60,126,255,255,126,60,255,255,251,127,255,247,125,222,251,191
1280 DATA0,12,6,255,255,255,6,12,153,219,126,60,60,126,219,153
1290 DATA255,255,195,195,195,195,255,255,30,59,63,28,28,22,22,22
1300 DATA22,114,158,156,152,152,24,30,0,7,31,56,96,199,207,255
1310 DATA254,255,1,0,254,255,255,255,0,192,240,56,12,198,230,254
1320 DATA24,24,24,24,24,24,24,24,60,60,24,126,189,60,66,129,-1
1330 REM=====MUSIKDATAS=====
1340 DATA834,834,739,770,798,810,834,834,739,739,739,739,854,854,810,834,854,873
1350 DATA881,881,739,739,739,739,810,810,834,810,798,770,798,798,810,798,770,739
1360 DATA722,722,739,770,798,739,770,770,770,770,834,834,739,770,798,810,834,834
1370 DATA739,739,739,739,854,854,810,834,854,873,881,881,739,739,739,739,810,810
1380 DATA834,810,798,770,798,798,810,798,770,739,770,770,798,770,739,722,739,739
1390 DATA739,739,-1
1400 DATA704,40,1020,5,704,40,1020,20,739,20,684,40,1020,5,684,40,1020,40,770,40
1410 DATA1020,5,770,40,1020,20,784,20,739,40,1020,5,739,40,1020,20,770,20,784,40
1420 DATA1020,5,784,40,1020,20,810,20,770,60,1020,5,770,20,834,20,822,20,834,20
1430 DATA854,20,864,40,854,20,834,20,1020,5,834,80,-1,-1
```

Ein Superspiel für den ZX-Spectrum (48k)
Otto Schweinsohr

Eine schreckliche Nachricht erschüttert die Softwarepiraten. Der gefürchtete Otto Schweinsohr, aufgebrochen von der Tafelrunde der heiligen H.A.C.K. ihre Untaten zu sühnen, kommt dem Kampfstern der Raubkopierer in der 17. Galaxie schon bedrohlich nahe. Natürlich schafft er es – aber nur wenn wir ihm helfen –. Doch wie immer gibt es Ärger, dafür sorgen:
– Der kleinliche Klaus und seine Clique
– Der bissige Berthold und seine Bande
– Der grimmige Gottlieb und seine Gang
Außerdem wird unser Held auch noch von einer bekannten staatlichen Institution verfolgt. Grund: Die interstellare Funkanlage aus Ottos „fliegender Kiste" besitzt keine FTZ-Prüfsumme.

Das Spiel selbst ist eine Routine, die aus 100% Maschinensprache besteht, die durch eine Art „Systemvariable" vom Basic aus gesteuert wird.
Beispiel: Poke 57615,x bestimmt die Zahl der Leben (auf keinen Fall mehr als 4 einpoken, sonst scrollt der Bildschirm). Für die müheselige Eintipparbeit entschädigt allein schon die aufwendige Grafik. Das Raumschiff, das euch in einer der höheren Spielstufen zur Verfügung steht, setzt sich aus 48 Grafikzeichen (!) zusammen.

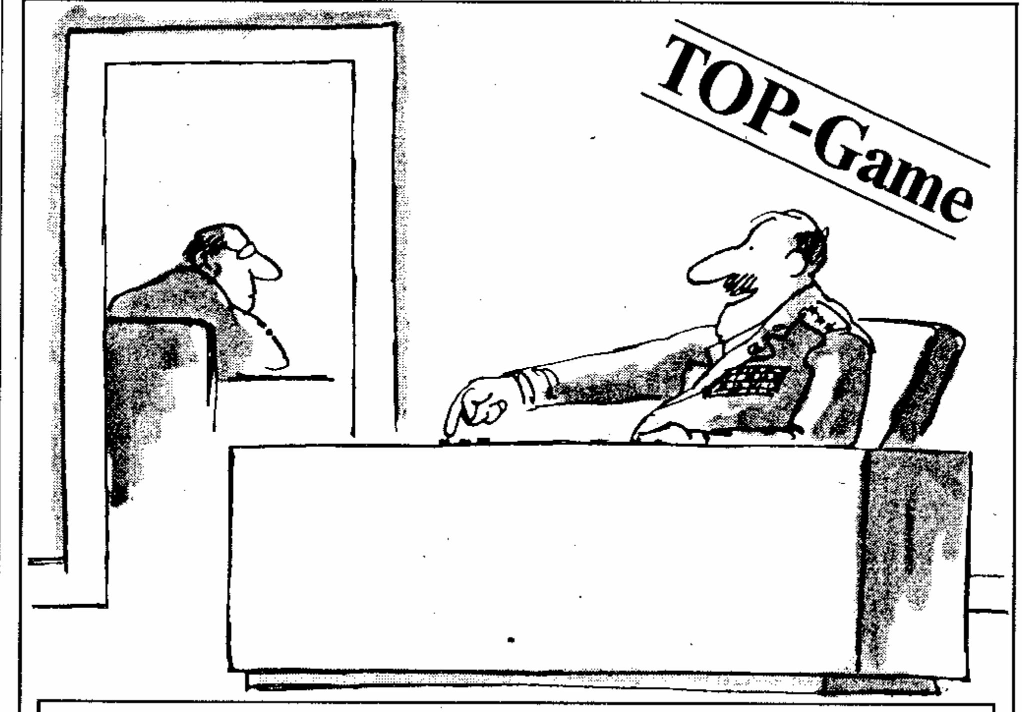
Zur Eingabe:
"LOAD" eingeben – das Programm wird geladen ebenso wie drei Maschinen-Codes und die Grafikteile.
Mit "Goto 9700" kann man die Codes auch per Hand eingeben.
Mit "Goto 9800" druckt man die Codes abdruckreif aus.

Einige Pokes:
57615,x :
Lebensanzahl (nicht über 4)
50001,x :
Länge der einzelnen Runden (16 Bit)
50002,x :
Länge der einzelnen Runden (16 Bit)
Insgesamt sind 3 x 3 = 9 Runden vorhanden.

Startadressen:
M-Code: Start 50.000 Länge 2300 Bytes
Grafik I : Start 58.880 Länge 4 Zeichensätze, also 4 x 768 = 3072 Bytes
Grafik II: ab „USR" „a" (also Grafikzeichen) bis Ende des Speichers ... → 65368 Länge, 168

Die Systemvariablen:

Inhalt	Adresse	Bedeutung
14	57600	Spalte – des 1. Alien (Feind)
14	57601	Zeile – des 1. Alien (Feind)
81	57602	Alienanzeige (ASCII-Code des Aliens, hier 81)
0	57603	2 Byteadressen des „Alienpanel" (Anzeige der Bewegungsart)
215	57604	
14	57605	Posistion des 2. Alien
14	57606	
32	57607	Alienpanel d. 2. Alien
215	57608	
14	57609	Position des 3. Alien
14	57610	
64	57611	Panel III
215	57612	
24 od. 17	57613	Limit (zeigt die erlaubte Flughöhe an (Zeile))
1–3	57614	Runde (wichtig für die Scorezählung)
4	57615	Leben
33	57616	Zeile des eigenen Raumschiffes
16	57617	Spalte des eigenen Raumschiffes
6	57618	Position der staatlichen Institution
12	57619	
	57620	2-Byte-Score Anfangs bei 00)
	57621	
	57622	2-Byte-Highscore (Anfangs bei 500)
	57623	



„Meyer, schaffen Sie mir auf der Stelle diesen Otto Schweinsohr hierher. Der kann in Zukunft mein Kaminholz hacken, aber keine Computer-Programme mehr!"

Otto Schweinsohr ist Super!

Listing – Teil Eins

```
   0>REM
     von CHRISTOPH HOLZHEUER

          © 4/1985

   2 GO SUB 9400
   3 DEF FN f(s)=PEEK s+256*PEEK
 (s+1)
  10 BORDER 0: PAPER 0
  15 CLS : LET za=4
  16 GO SUB 8100
  17 PLOT 36,112
  18 GO SUB 8150
  20 LET za=12: INK 3: GO SUB 82
 00
  21 PLOT 88,112: LET za=8
  22 GO SUB 8200: PLOT 104,128
  23 LET za=4: GO SUB 8200
  24 PRINT AT 20,1;" *** ~SPACE~
 : Hauptmenu *** "
  25 PLOT 164,112: LET za=1
  26 GO SUB 8200: DRAW 0,40,-3
  27 LET za=3: GO SUB 8200
  30 DRAW 0,-8,4: LET za=2
  32 GO SUB 8200: PLOT 176,112
  34 LET za=12: GO SUB 8200
  45 CIRCLE 82,115,3
  46 RANDOMIZE USR 50260
  47 CIRCLE 132,115,3
  48 RANDOMIZE USR 50260
  49 CIRCLE 169,115,3
  50 RANDOMIZE USR 50260
  52 CIRCLE 221,115,3
  53 RANDOMIZE USR 50260
  62 PLOT 110,22: GO SUB 8200
  63 PLOT 122,52: LET za=4
  64 GO SUB 8200
  65 INK 0: PAPER 1
  66 BEEP .7,-6: BEEP .9,-13: BE
 EP .6,-10
  67 PRINT AT 10,1;"[illegible]
 [illegible]"
  69 INK 7
  70 PRINT AT 11,1;"[illegible] "
  71 PRINT AT 12,1;"[illegible] "
  72 PRINT AT 13,1;"[illegible] "
  80 PRINT AT 11,10;"[illegible]
 [illegible]"
  81 PRINT AT 12,10;"[illegible]
 [illegible]"
  82 PRINT AT 13,10;"[illegible]
 [illegible]"
  83 PRINT #1;"* Anleitung erwue
 nscht?  (j/n) *"
  84 BRIGHT 1
  85 PRINT AT 9,10;"praesentiert
 "
  86 PAPER 0
  87 PRINT AT 15,21;"DESIGN:"
  88 PRINT AT 16,3;"M. Geiger"
  89 PRINT AT 16,19;"C.Holzheuer
 "
  90 PRINT AT 15,4;"GRAPHIK:"
  91 PRINT AT 20,2;"Copyright ©
 1985 by H.A.C.K."
  97 PAUSE 0
  98 IF INKEY$="n" THEN GO TO 30
 0
  99 REM Anleitung brav Abtippen
 100 CLS : INK 7: PRINT ''
 101 RANDOMIZE USR 50260
 105 PRINT "         Die Vorgeschi
 chte"
 106 PRINT "         -------------
 ----"
 107 PRINT ''
 108 INK 3
 110 PRINT "  Eine schreckliche
 Nachricht"
 111 PRINT "  erschuettert die S
 oftware-"
 112 PRINT "  piraten:"
 115 PRINT '"          Der gefuerch
 tete Otto"
 117 PRINT "  Schweinsohr, aufge
 brochen"
 120 PRINT "  von der Tafelrunde
 der heil-"
 122 PRINT "  igen H.A.C.K.  ihr
 e Untaten"
 124 PRINT "  zu suehnen,"
 126 PRINT "              kommt d
 em Kampf-"
 130 PRINT "  stern der Raubkopi
 erer in"
 132 PRINT "  der 17. Galaxie sc
 hon be-"
 134 PRINT "  drohlich nahe ..."
 136 GO SUB 8100: INK 3
 137 PRINT #1;"         **   TASTE
 ! **"
 140 PAUSE 0: CLS
 141 RANDOMIZE USR 50260
 150 PRINT ''"  Natuerlich schaf
 ft er es."
 152 PRINT "  (Aber nur wenn ";
 FLASH 1;"DU"; FLASH 0;" ihm hilf
 st)"
 154 PRINT "  Aber wie immer gib
 t es "
 156 PRINT "  Aerger:"
 160 PRINT "            Der Aerger
 :"
 162 PRINT AT 10,0
 164 PRINT "  Der       Der
   Der "
 168 PRINT "  klein-    bissige
  grimmige"
 170 PRINT "  liche     Bertold
  Gottlieb"
 172 PRINT "  Klaus       und
   und"
 174 PRINT "   und        seine
   seine"
 176 PRINT "  seine     Bande
   Gang"
 177 PRINT "  Clique": INK 7
 178 GO SUB 8000: POKE 57603,0
 179 POKE 23607,232
 180 POKE 57600,29
 181 POKE 57601,16
 182 POKE 57605,8: POKE 57607,0
 183 POKE 57606,16: POKE 57611,0
 184 POKE 57609,18
 185 POKE 57610,16
 187 PRINT ;: LET l=USR 50432
 188 POKE 57602,85
 189 PRINT ;: LET l=USR 50485
 190 POKE 57602,89
 191 PRINT ;: LET l=USR 50541
 192 POKE 23607,60: INK 3
 193 PRINT #1;"         **   TASTE
 ! **"
 195 GO SUB 8100: INK 3
 199 PAUSE 0
 200 CLS : RANDOMIZE USR 50260
 201 PRINT ''"  Ausserdem  wird
 unser  Held"
 203 PRINT "  auch noch vom Bund
 espostmi-"
 205 PRINT "    nister (Oho!) ve
 rfolgt!"
 207 PRINT '"  Grund: Die Inters
 tellare"
 209 PRINT "    Funkanlage aus O
 ttos"
 211 PRINT TAB 7;"Bomber hat kei
 ne"
 213 PRINT "      FTZ Pruefnummer
 ...."
 240 PRINT AT 16,0
 242 PRINT "  Bit und Bytebruch
 wuenscht"
 250 PRINT '"      euch eure   H.
 A.C.K.  !"
 251 GO SUB 8100
 260 POKE 57619,9
 261 FOR n=9 TO 21
 262 POKE 57618,n
 263 RANDOMIZE USR 51345
 270 NEXT n
 271 REM das war der POSTMINISTA
 280 PRINT #1;"         **   TASTE
 ! **"
 291 INK 7
 299 PAUSE 0
 300 INK 5: CLS
```


IMPRESSUM

Computronic
erscheint alle 2 Monate im Tronic-Verlag, 3444 Wehretal 1
Tel. 0 56 51 / 4 06 93 oder 4 06 43
3. Jahrgang

Herausgeber: Axel Credé
Chefredakteure: Siegfried Görk/Bernd Zimmermann
Redaktion: Manfred Kleimann, Hartmut Wendt, Frank Brall, Ottfried Schmidt

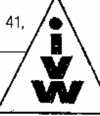
Programmautoren: D. Bludau, U. Brehmer, B. Dallé, S. Ghaffari, E. u. J. Heß, C. Holzheuer, J. Klose, R. Mayer, H. Ott, C. Raskob, U. Sigmund, O. Steinmeier, O. Thimm, O. Windel, E. Wittig, M. Wolfgram

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 0 61 21 / 26 60

Anzeigenverwaltung (Inland):
Hartmut Wendt, Tronic-Verlag, Postfach 41, 3444 Wehretal 1, Tel. 0 56 51 / 4 06 93

Anzeigenleitung (Ausland):
Public Relation (Foreign Countries):
Manfred Kleimann, c/o Tronic-Verlag,



Postfach 41 (Post Box 41)
D-3444 Wehretal 1, Tel. 0 56 51 / 4 06 43

Erscheinungsweise: Erstverkaufstag von COMPUTRONIC ist Anfang des Monats

Anzeigenpreisliste:
Bitte Mediaunterlagen anfordern.

Software-Service und Vertrieb: Annelie Kratzenberg, Heike Rabe

Bezugspreise:
Einzelheft 6,50 DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen. Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,– DM pro abgedruckter Seite im Heft einverstanden sind.

Titelblatt: Werbeatelier H. Kästle, Eschwege

```
 310 PRINT '"   Was darf's denn
heute sein ?"
 311 PRINT
 313 PRINT "                MENU"
 315 PRINT "                ----"
 320 PRINT '"      [1]  Kempston
Joystick"
 322 PRINT '"      [2]  Tasten: (
1,q,~ ~)"
 324 PRINT '"      [3]  Raus dami
t! (NEW)"
 327 PRINT '"      [4]  Auf ins A
benteuer"
 330 PRINT '''"         © 1985 by
H.A.C.K."
 333 PRINT '"  FSK: freigegeben
ab 3 Jahren"
 335 GO SUB 8100: INK 5
 350 LET q=CODE INKEY$
 351 IF q>52 THEN GO TO 350
 352 IF q<49 THEN GO TO 350
 353 BEEP .3,3
 356 GO TO 360+5*(q-48)
 360 GO TO 350
 365 POKE 22758,199
 366 POKE 50054,96
 367 POKE 50055,198
 368 GO TO 350
 369 REM das war fuern Joystick
 370 POKE 22822,199
 371 POKE 50054,238
 372 POKE 50055,202
 373 GO TO 350
 374 REM das war fuers Keyboard
 375 STOP
 380 REM ab ins spiel
 399 REM ab 400:Vorbereitung
 400 POKE 57615,4
 401 POKE 57620,0: POKE 57621,0
 402 POKE 50001,124
 403 POKE 50002,1
 409 INK 7
 410 GO SUB 8000
 500 FOR x=1 TO 3
 501 POKE 57614,x
 502 POKE 57613,24
 503 IF x=3 THEN GO SUB 8050
 510 GO SUB 6000
 700 FOR Y=1 TO 3
 701 CLS
 702 INK 4+x
 705 POKE 23607,(x*3)+226
 706 GO SUB 4890+(x*100)+(y*10)
 707 LET l=USR 50841: REM score
 710 FOR n=14 TO 16
 711 POKE 57617,n: REM zeile
 712 LET l=USR 50150
 713 LET l=USR 50713
 714 IF x=3 THEN LET l=USR 52220
 720 BEEP .2,n: NEXT n
 725 BEEP .2,-2: BEEP .4,0
 800 LET l=USR 50000
 801 REM das war's HAUPTPROGRAM
 850 LET w=PEEK 57617: REM zeile
 851 FOR n=w TO 14 STEP -1
 852 POKE 57617,n: REM zeile
 853 LET l=USR 50150
 854 LET l=USR 50713
 855 IF x=3 THEN LET l=USR 52220
 856 BEEP .2,n: NEXT n
 860 BEEP .2,0: BEEP .4,-2
 861 POKE 23607,60
 862 INK 3
 865 IF PEEK 23560=121 THEN GO T
O 885
 866 PRINT AT 10,10;"Gelandet!"
 868 PRINT #1; FLASH 1;"
**  TASTE! **",: PAUSE 0
 869 PRINT AT 10,10;"          "
 870 IF PEEK 57615>0 THEN NEXT y
 871 IF PEEK 57615>0 THEN NEXT x
 873 IF x=4 THEN GO SUB 7e3
 880 PRINT AT 8,12;"GAME OVER"
 885 LET s=57620: REM score
 886 LET p=FN f(s)
 887 LET s=57622: REM hiscore
 888 LET hi=FN f(s)
 889 IF p>hi THEN BEEP 1,1: INPU
T "Dein werter Name:";h$: POKE 5
7622,PEEK 57620: POKE 57623,PEEK
 57621: LET hi=FN f(s)
 890 PRINT AT 10,11;"Score: 0";p
 891 PRINT AT 12,13;"Hi: ";hi
 892 PRINT AT 14,11;"von: ";h$
 900 PLOT 78,116
 901 DRAW 109,0: DRAW 0,-70
 902 DRAW -109,0: DRAW 0,70
 910 PAUSE 0
 911 BEEP 2,-5
 998 RESTORE : GO TO 10
5000 REM landschaft 1.1
5001 PRINT AT 16,0; INK 4;"      i
j       ij      gh  ij      gh    k
l       kl      ij   kl      ij    m
n ij    mn      kl   mn      kl
  kl            mn           mn
  mn"
5002 PLOT 0,23
5005 DRAW 255,0
5006 RETURN
5010 REM landschaft 1.2
5011 PRINT AT 16,0; INK 2;"
op        op        op          o
qqpop  oqqp  op  oqqp  op      or
sqrqqpoqrsqpoqqpoqrsqpoqqp   orq
qsqqqrqrqqqrqqqrqrqqsqsqqqp"
5013 RETURN
5020 REM landschaft 2.1
5021 PRINT AT 16,0; INK 1;"ututu
tutututututututututututututvwvwv
wvwvwvwvwvwvwvwvwvwvwvwvwvwvwvwv
wvwvwvwvwvwvwvwvwvwvwvwvwvwvwvwv
wvwvwvwvwvwvwvwvwvwvwvwvwvwvwvwv
wvwvwvwvwvwvwvwvwvwvwvwvwvwvwvwv
wvwvwvwvwvwvwvwvwvwvwvwvw"
5024 RETURN
5100 REM landschaft 2.1
5101 PRINT AT 16,0; INK 6;"jkl██
█jkl██hl█jkl██hi██jkl█jkl█hi█hi█
hi██jkl██hi██jkl██hi████jkl███jk
l██hi█m██ji█jkl hi██████hi███jkl█
██jkl██jl████jkl███m██jl██████hi
███hi████kl████m███jkl██jkl███m██
m██m██hi██████m█████hi███jkl████"
5105 RETURN
5110 REM landschaft 2.2
5111 INK 4: PRINT AT 16,0;"no
      no          no           pq
 no  pq       pq       no      rs
 pq  rs no   rs       pq    no
 rs      pq           rs    pq
         rs                 rs"
5112 PLOT 0,23: DRAW 255,0
5113 INK 6: RETURN
5120 REM landschaft 2.3
5121 PLOT 0,18
5122 DRAW 60,0,-3: DRAW 120,-10,
-1
5124 DRAW 75,5,1
5125 PLOT 50,36
5127 DRAW 60,-6,-1.7
5130 RETURN
5200 REM landschaft 3.1
5201 PRINT AT 16,0; INK 5;"ghjik
ljikhhjhikhhhjhjhjhjhjghghgghjik
lghijklghijklghijklghjiklghjiklg
ghjiklghjiklghjiklghjiklghjiklgh
jiklghjiklghjikllghjiklghjiklghj
ikkghjiklghjiklghjkikghjiklhijkl
jihjklikljijhjkijjkjhhjkiki"
5202 PRINT AT 0,0; INK 5;"ghjikl
jikhhjhikhhhjhjhjhjhjghghgghjikl
ghijklghijklghijklghjiklghjiklgg
hjiklghjiklghjiklghjiklghjiklghj
iklghjiklghjikllghjiklghjiklghji
kkghjiklghjiklghjkikghjiklnropqn
ropqnropqnropqnropqnropqnr"
5209 RETURN
5210 REM landschaft 3.2
5211 PRINT AT 16,0; INK 2;"mtutu
utuutuutuutututtuuutuutuutuututu
uutututututuuttuttuuttutututtutt
ututtuttutttutuutuututtutuuutuut
utuutuututututututuututututtutut
ututututututututututututuuututuu
tutututuutuutuuttuuuttuutt"
5212 PRINT AT 0,0; INK 3;"vwx
 vwx     vwx        vwx          yz(
 yz(    yz( vwx  yz(   vwx
             yz(        yz(       vwx
      vwx                         yz(
      yz(    vwx      vwx
             yz(      yz("
5214 RETURN
5220 REM landschaft 3.3
5221 FOR n=0 TO 255 STEP 16
5223 PLOT n,0: DRAW 0,47
5224 NEXT n
5226 FOR n=1 TO 7
5244 PLOT 0,48-(n^2): DRAW 255,0
5247 NEXT n
5249 FOR n=0 TO 255 STEP 16
5250 PLOT n,175: DRAW 0,-46
5251 NEXT n
5252 FOR n=7 TO 2 STEP -1
5253 PLOT 0,(n^2)+125: DRAW 255,
0
5255 NEXT n
5266 INK 7
5299 RETURN
5999 REM das ist der Vorspann
6000 CLS : RESTORE 9040
6001 PAPER 1: PRINT AT 15,1
6002 LET l=2
6003 FOR n=16 TO 20
6004 IF n=17 THEN PAPER 3
6005 PRINT AT n,1;"
                              "
6007 NEXT n
6015 GO SUB 8100: INK 7: PAPER 0
6016 PRINT AT 2,12;x;". Runde"
6020 PRINT AT 13,22;"AIRPORT"
6021 PLOT 174,62
6022 DRAW 56,0: DRAW 6,6: DRAW -
6,6: DRAW -56,0: DRAW 0,-12
6023 PLOT 197,63
6024 DRAW 0,-20: DRAW 3,0: DRAW
0,20
6025 PLOT 8,37: DRAW 239,0
6026 PLOT 8,34: DRAW 239,0
6027 PLOT 36,130
6028 DRAW 87,0: DRAW 0,-12
6029 DRAW -87,0: DRAW 0,12
6030 INK 7
6031 DIM a$(6,3)
6040 LET a$(1)=" UV"
6041 LET a$(2)=" WX"
6042 LET a$(3)=" ]^"
6043 LET a$(4)=" _£"
6044 LET a$(5)=" ab"
6045 LET a$(6)=" cd"
6047 LET l=1
6048 FOR f=1 TO 6
6049 FOR n=1 TO 6 STEP 2
6050 POKE 23607,238
6051 PRINT AT 12,1;" QR"
6052 PRINT AT 13,1;" ST"
6057 PRINT AT 14,1;a$(n)
6058 PRINT AT 15,1;a$(n+1)
6060 READ a,b: BEEP a,b-10
```

```
6061 LET l=l+1
6063 POKE 23607,60
6065 PRINT AT 6,5;b$(x,l TO l+9)
6070 NEXT n
6071 NEXT f
6075 POKE 23607,60
6079 PAUSE 60
6080 RETURN
7000 CLS
7009 PRINT '"      G R A T U L A
 T I O N  "
7010 PRINT '"  Naja,fuer den An
fang nicht"
7020 PRINT "  allzu schlecht.Du
 hast die"
7030 PRINT "  Sperranlagen beme
rkenswert"
7040 PRINT "  gut ueberstanden.
"
7050 PRINT '"  Leider musst Du
aber ent-"
7060 PRINT "  decken,dass der K
ampfstern"
7070 PRINT "  seit 20 Jahren to
tal ver-"
7080 PRINT "  lassen vor sich h
in schim-"
7090 PRINT "  melt."
7100 PRINT "  Wie's weitergeht
?"
7110 PRINT '"  Gute Frage! Die
Antwort "
7120 PRINT "  findest Du in dem
 Nachfolge-"
7130 PRINT "  spiel OTTO SCHWEI
NSOHR II"
7140 GO SUB 8100: PAUSE 0
7150 CLS : RETURN
7699 REM on break goto
7700 PRINT AT 10,2;"Na Na Na! Bi
tte kein Break!"
7701 BEEP 1,0: PAUSE 50
7710 PRINT AT 10,2;"
                  "
7720 RETURN
7999 REM systemvars
8000 POKE 57600,4: POKE 57601,14
8001 POKE 57602,81
8002 POKE 57603,0
8003 POKE 57604,215
8005 POKE 57605,6: POKE 57606,16
8007 POKE 57607,32
8008 POKE 57608,215
8010 POKE 57609,8: POKE 57610,18
8011 POKE 57611,64
8012 POKE 57612,215
8015 POKE 57616,33
8016 POKE 57617,14
8020 POKE 57618,8: POKE 57619,12
8025 POKE 57620,0: POKE 57621,0
8026 POKE 57627,13
8027 POKE 57626,8
8028 POKE 57628,0
8030 RETURN
8049 REM letzter level
8050 GO SUB 8000
8051 POKE 57613,18: REM limit
8052 IF PEEK 57615<4 THEN POKE 5
7615,(PEEK 57615)+1
8053 POKE 50002,0
8054 POKE 50001,190
8060 RETURN
8100 INK 1: DRAW 255,0: DRAW 0,1
75: DRAW -255,0: DRAW 0,-175
8101 PLOT 5,5
8102 DRAW 245,0: DRAW 0,165: DRA
W -245,0: DRAW 0,-165
8105 RETURN
8150 INK 7
8155 FOR f=1 TO 4
8160 READ w$,T
8165 FOR n=19 TO 1 STEP -1
8170 PRINT AT n,T;w$;AT n+1,t;"
   "
8171 NEXT n
8175 RANDOMIZE USR 50210
8180 NEXT f: RETURN
8200 FOR n=1 TO za: READ x,y
8205 DRAW x,y: NEXT n
8206 IF INKEY$=" " THEN GO TO 30
0
8210 RETURN
8999 STOP
9002 DATA "HYDRO",2,"AKTIVE",8,"
COMPUTER",15," KOMUNE",23
9005 DATA 0,40,16,0,0,-16,8,0,0,
16,16,0,0,-40,-16,0,0,16,-8,0,0,
-16,-16,0
9010 DATA 16,40,8,0,16,-40,-16,0
,0,8,-8,0,0,-8,-16,0,8,0,0,8,-8,
0,0,-8
9020 DATA -13,0,13,0,0,-16,-8,0,
8,0,0,-16
9030 DATA 0,40,16,0,0,-16,24,16,
0,-16,-8,-6,8,-18,-16,0,-5,10,-3
,-2,0,-8,-16,0
9035 DATA 0,15,8,0,0,4,6,0,0,8,7
,0,0,-8,6,0,0,-4,8,0,0,-15,-34,0
9037 DATA 11,0,0,7,-11,0,0,-7
9040 DATA .5,20,.25,22,.125,20,.
375,17,.5,17,.25,17,.125,15,.25,
17,.125,18,.5,17,.5,18,.25,15,.1
25,20,.75,17,.5,13,.25,10,.125,1
5,.75,8
9050 RETURN
9400 IF PEEK 5e4<>0 THEN GO TO 9
500
9410 LOAD ""CODE
9411 LOAD ""CODE
9412 LOAD ""CODE
9500 POKE 57622,244: POKE 57623,
1
9501 LET a$=" Score:      Hi:
 Ru:"
9502 LET h$="EMPIRE"
9510 FOR n=1 TO LEN a$
9511 POKE n+55166,CODE a$(n)
9512 NEXT n
9520 DIM b$(3,50)
9521 LET b$(1)=" Otto startet vo
n der Erde"
9522 LET b$(2)=" Die Raumbase is
t erreicht"
9523 LET b$(3)=" Wir sind am Kam
pfstern..."
9599 RESTORE 9605
9600 FOR n=55040 TO 55071: READ
a: POKE n,a: NEXT n
9601 FOR n=55072 TO 55103: READ
a: POKE n,a: NEXT n
9602 FOR n=55104 TO 55135: READ
a: POKE n,a: NEXT n
9603 RESTORE
9605 DATA 0,0,1,0,0,255,0,1
9606 DATA 0,0,1,0,0,255,0,1
9607 DATA 0,0,1,0,0,255,0,1
9608 DATA 0,0,1,0,0,255,0,1
9610 DATA 1,255,1,255,0,0,0,1
9611 DATA 1,255,1,255,0,0,0,1
9612 DATA 1,255,1,255,0,0,0,1
9613 DATA 1,255,1,255,0,0,0,1
9620 DATA 0,1,0,255,0,1,0,0
9621 DATA 0,1,0,255,0,1,0,0
9622 DATA 0,1,0,255,0,1,0,0
9623 DATA 0,1,0,255,0,1,0,0
9630>REM das war die Alienbewegu
ng
9699 RETURN
9700 LET a=5e4
9710 INPUT i
9720 PRINT a,i: POKE a,i
9730 LET a=a+1
9740 IF a=61697 THEN LET a=USR "
a"
9741 IF a=52195 THEN LET a=58880
9750 GO TO 9710
9799 STOP
9800 LET x=5e4
9810 PRINT x;
9820 FOR n=6 TO 29 STEP 4
9830 PRINT TAB n;PEEK x;
9835 LET x=x+1
9840 IF x=52196 THEN LET x=58880
9841 IF x=61700 THEN LET x=USR "
a"
9845 NEXT n: PRINT
9850 GO TO 9810
9990 CLEAR
9991 SAVE "OTTO  I" LINE 1
9992 SAVE "OTTOMC"CODE 5e4,2300
9993 SAVE "OTTOGR"CODE 58880,4*7
68
9994 SAVE "OTTOPM"CODE USR "a",7
68
9999 GO TO 9991
```

Listing: Teil Zwei

```
50000 17  124 1   213 58  14
50006 225 254 1   32  8   205
50012 53  197 205 230 195 24
50018 30  254 2   32  11  205
50024 53  197 205 0   197 205
50030 230 195 24  15  205 53
50036 197 205 0   197 205 109
50042 197 205 230 195 205 252
50048 203 58  17  225 245 205
50054 238 202 237 75  16  225
50060 62  13  184 32  1   4
50066 58  13  225 184 32  1
50072 5   237 67  16  225 241
50078 184 196 25  198 205 6
50084 199 205 207 201 209 213
50090 123 254 3   32  5   62
50096 111 50  28  225 205 32
50102 203 0   58  28  225 254
50108 111 204 145 200 58  18
50114 225 254 8   32  5   62
50120 0   50  28  225 209 58
50126 15  225 254 0   200 58
50132 8   92  254 121 200 27
50138 62  0   186 194 83  195
50144 187 194 83  195 201 0
50150 33  0   80  84  93  27
50156 6   8   197 6   6   197
50162 1   31  0   19  35  26
50168 237 176 18  193 16  243
50174 1   64  0   9   84  93
50180 27  193 16  230 201 6
50186 64  17  0   64  213 225
50192 35  197 1   31  0   26
50198 237 176 43  119 0   35
50204 35  19  193 16  240 201
50210 17  60  60  1   0   8
50216 33  52  4   58  72  92
50222 203 63  203 63  203 63
50228 197 203 231 211 254 66
50234 21  24  0   16  252 70
50240 16  254 203 167 211 254
```

50246	67	0	24	0	16	252
50252	70	16	254	35	193	16
50258	225	201	17	45	80	1
50264	0	30	33	52	4	58
50270	72	92	203	63	203	63
50276	203	63	197	203	231	211
50282	254	66	20	24	0	16
50288	252	70	16	254	203	167
50294	211	254	67	20	24	0
50300	16	252	70	16	254	0
50306	193	16	225	201	17	45
50312	45	1	0	33	33	52
50318	4	58	72	92	203	63
50324	203	63	203	63	197	203
50330	231	211	254	66	29	24
50336	0	16	252	70	16	254
50342	203	167	211	254	67	29
50348	24	0	16	252	70	16
50354	254	35	193	16	225	201
50360	205	124	0	59	59	225
50366	1	15	0	9	235	42
50372	61	92	115	35	114	201
50378	118	205	142	2	123	254
50384	255	32	248	58	58	92
50390	254	12	40	10	254	16
50396	40	6	254	20	40	2
50402	24	25	60	50	129	92
50408	253	54	0	255	33	2
50414	0	34	66	92	33	0

50636	5	205	217	13	193	241
50642	60	245	215	241	60	215
50648	62	128	215	62	6	184
50654	200	197	5	5	205	217
50660	13	62	128	215	62	128
50666	215	62	128	215	193	201
50672	197	205	217	13	62	128
50678	215	62	128	215	193	197
50684	5	205	217	13	62	128
50690	215	62	128	215	6	4
50696	58	2	225	128	254	93
50702	32	2	62	81	50	2
50708	225	193	14	3	201	62
50714	2	205	1	22	237	75
50720	16	225	197	205	217	13
50726	62	33	6	6	197	6
50732	8	245	215	241	60	16
50738	250	245	62	13	215	241
50744	193	16	239	193	58	13
50750	225	197	184	40	4	4
50756	205	85	198	193	62	14
50762	184	200	120	6	6	144
50768	71	205	85	198	201	205
50774	217	13	6	8	62	128
50780	215	16	251	201	14	31
50786	46	0	237	120	237	75
50792	16	225	254	4	32	1
50798	5	254	8	32	1	4
50804	254	24	32	3	4	38

51026	50	26	225	62	16	215
51032	62	7	215	4	205	56
51038	37	205	241	43	26	254
51044	32	200	58	26	225	71
51050	62	18	144	14	14	129
51056	237	75	0	225	33	0
51062	225	185	40	35	60	185
51068	40	31	237	75	5	225
51074	33	5	225	185	40	21
51080	61	185	40	17	237	75
51086	9	225	33	9	225	185
51092	40	7	60	185	40	3
51098	195	26	200	0	22	3
51104	229	213	197	62	16	215
51110	62	2	215	205	217	13
51116	62	92	215	62	93	215
51122	193	197	5	205	217	13
51128	62	94	215	62	95	215
51134	205	134	196	62	16	215
51140	62	7	215	193	197	205
51146	217	13	62	96	215	62
51152	97	215	193	197	5	205
51158	217	13	62	98	215	62
51164	99	215	205	134	196	193
51170	197	205	217	13	62	128
51176	215	62	128	215	193	197
51182	5	205	217	13	62	128
51188	215	62	128	215	193	209
51194	21	62	0	186	32	161

Nur nicht aufgeben! – Eure Bemühungen lohnen sich garantiert!!

50420	0	34	68	92	59	59
50426	195	125	27	195	3	19
50432	237	75	9	225	12	62
50438	33	185	32	6	13	205
50444	240	197	14	4	58	13
50450	225	184	32	6	13	205
50456	240	197	6	12	42	11
50462	225	126	128	71	35	62
50468	95	189	32	2	46	64
50474	34	11	225	237	67	9
50480	225	205	165	197	201	205
50486	102	203	237	75	0	225
50492	12	62	33	185	32	6
50498	13	205	240	197	14	4
50504	58	13	225	184	32	6
50510	13	205	240	197	6	12
50516	42	3	225	126	128	71
50522	35	62	95	189	32	2
50528	46	64	34	3	225	237
50534	67	0	225	205	165	197
50540	201	237	75	5	225	12
50546	62	33	185	32	6	13
50552	205	240	197	14	4	58
50558	13	225	184	32	6	13
50564	205	240	197	6	12	42
50570	7	225	126	128	71	35
50576	62	95	189	32	2	46
50582	64	34	7	225	237	67
50588	5	225	205	165	197	201
50594	0	0	0	58	13	225
50600	184	40	15	197	4	205
50606	217	13	62	128	215	62
50612	128	215	62	128	215	193
50618	197	205	217	13	193	58
50624	2	225	245	215	241	60
50630	245	215	62	128	215	197

50810	111	254	20	32	3	5
50816	38	111	254	16	32	2
50822	38	111	237	67	16	225
50828	62	111	188	192	34	24
50834	225	201	0	0	0	0
50840	0	62	2	205	1	22
50846	62	0	50	107	92	1
50852	33	2	205	217	13	58
50858	15	225	254	0	40	9
50864	71	62	101	215	205	227
50870	203	16	248	58	55	92
50876	245	62	60	50	55	92
50882	1	25	2	205	217	13
50888	1	24	0	17	127	215
50894	205	60	32	1	18	2
50900	205	217	13	237	75	20
50906	225	205	27	26	1	10
50912	2	205	217	13	237	75
50918	22	225	205	27	26	1
50924	2	2	205	217	13	237
50930	75	14	225	6	0	205
50936	27	26	241	50	55	92
50942	62	2	50	107	92	201
50948	92	201	58	25	225	254
50954	0	200	58	26	225	254
50960	8	32	11	237	75	16
50966	225	62	24	144	60	50
50972	27	225	205	34	196	62
50978	2	205	1	22	62	16
50984	215	62	5	215	62	22
50990	215	58	27	225	79	215
50996	58	26	225	71	215	62
51002	128	215	62	127	215	4
51008	62	18	184	32	12	62
51014	0	50	25	225	6	8
51020	120	215	62	128	215	120

51200	14	4	225	113	42	20
51206	225	58	14	225	17	6
51212	0	25	61	254	0	32
51218	250	34	20	225	205	153
51224	198	201	62	21	215	62
51230	1	215	62	16	215	62
51236	4	215	205	113	200	62
51242	16	215	62	2	215	205
51248	113	200	62	16	215	62
51254	5	215	205	113	200	62
51260	21	215	62	0	215	205
51266	113	200	62	16	215	62
51272	7	215	205	113	200	62
51278	21	215	62	1	215	205
51284	113	200	62	21	215	62
51290	0	215	17	111	0	42
51296	20	225	25	34	20	225
51302	205	153	198	205	218	202
51308	201	0	0	0	0	237
51314	75	18	225	13	62	5
51320	245	197	205	217	13	6
51326	6	62	79	215	16	251
51332	193	241	4	61	254	0
51338	32	236	205	134	196	201
51344	0	62	2	205	1	22
51350	237	75	18	225	205	217
51356	13	62	143	245	6	6
51362	62	16	215	62	6	215
51368	241	60	245	215	16	250
51374	62	128	215	237	75	18
51380	225	4	205	217	13	6
51386	6	62	16	215	62	6
51392	215	241	60	245	215	16
51398	250	62	128	215	237	75
51404	18	225	4	4	13	13
51410	205	217	13	6	4	62

51416	16	215	62	6	215	241
51422	60	245	215	16	250	62
51428	128	215	237	75	18	225
51434	4	4	4	13	13	13
51440	205	217	13	6	3	62
51446	16	215	62	6	215	241
51452	60	245	215	16	250	62
51458	128	215	237	75	18	225
51464	62	4	128	71	62	252
51470	129	79	205	217	13	62
51476	18	215	62	1	215	6
51482	1	62	16	215	62	2
51488	215	241	60	0	215	16
51494	250	62	18	215	62	0
51500	215	62	128	215	205	84
51506	196	237	75	18	225	12
51512	62	23	185	40	5	237
51518	67	18	225	201	42	16
51524	225	62	24	148	103	62
51530	18	145	6	16	128	0
51536	0	71	14	10	62	16
51542	215	62	5	215	62	22
51548	215	121	215	120	215	62
51554	164	215	62	128	215	229
51560	197	33	220	5	17	15
51566	0	205	181	3	193	197
51572	5	205	56	37	205	241
51578	43	193	225	26	254	32
51584	32	64	121	188	40	10
51590	13	62	8	215	62	8
51596	215	62	128	215	16	200
51602	62	8	215	62	8	215
51608	62	128	215	62	16	215
51614	62	7	215	0	0	237
51620	75	18	225	12	237	67
51626	18	225	205	26	200	42
51632	20	225	17	111	0	237
51638	82	34	20	225	1	8
51644	12	237	67	18	225	201
51650	62	3	184	32	201	205
51656	146	201	205	30	202	201
51662	0	237	75	0	225	62
51668	26	185	32	13	72	6
51674	6	58	17	225	185	202
51680	30	202	61	16	249	237
51686	75	5	225	62	26	185
51692	32	13	72	6	6	58
51698	17	225	185	202	30	202
51704	61	16	249	237	75	9
51710	225	62	26	185	192	72
51716	6	6	58	17	225	185
51722	202	30	202	61	16	249
51728	201	185	192	72	6	6
51734	241	185	40	4	61	16
51740	250	201	58	15	225	61
51746	50	15	225	62	2	205
51752	1	22	62	21	215	62
51758	1	215	62	16	215	62
51764	2	215	205	177	202	62
51770	21	215	62	0	215	62
51776	16	215	62	1	215	205
51782	177	202	62	21	215	62
51788	1	215	205	177	202	62
51794	21	215	62	0	215	58
51800	55	92	245	62	238	50
51806	55	92	17	30	0	33
51812	0	15	213	229	205	181
51818	3	205	25	198	205	230
51824	195	205	208	202	225	36
51830	209	237	75	16	225	62
51836	14	184	40	7	5	237
51842	67	16	225	24	223	0
51848	241	50	55	92	62	21
51854	215	62	1	215	205	25
51860	198	62	21	215	62	0
51866	215	62	16	215	62	7
51872	215	205	25	198	17	0
51878	1	33	0	5	205	181
51884	3	205	153	198	201	237
51890	75	16	225	205	217	13
51896	6	6	197	6	8	62
51902	88	215	16	251	62	13
51908	215	193	16	242	205	134
51914	196	201	0	0	0	0
51920	58	14	225	254	3	204
51926	252	203	201	0	1	8
51932	12	237	67	18	225	62
51938	0	50	28	225	201	0
51944	0	0	0	0	0	0
51950	237	91	16	225	42	24
51956	225	1	254	247	237	120
51962	203	71	32	1	20	1
51968	254	251	237	120	203	71
51974	32	1	21	1	254	127
51980	237	120	203	71	32	2
51986	38	111	34	24	225	237
51992	83	16	225	201	0	0
51998	0	0	17	0	0	237
52004	107	17	225	38	0	41
52010	41	205	181	3	33	0
52016	225	126	205	66	203	17
52022	5	0	25	126	205	66
52028	203	17	4	0	25	126
52034	6	8	14	33	185	40
52040	4	13	16	250	201	35
52046	126	237	75	16	225	72
52052	6	6	185	202	25	198
52058	13	16	249	201	0	0
52064	0	0	0	0	0	0
52070	58	120	92	254	75	40
52076	7	254	150	40	3	254
52082	225	192	237	75	0	225
52088	62	31	185	200	60	185
52094	200	60	185	200	62	18
52100	145	22	15	130	72	71
52106	62	14	145	22	10	130
52112	79	5	5	62	16	215
52118	62	5	215	62	22	215
52124	121	215	120	215	62	127
52130	215	62	128	215	197	5
52136	205	56	37	205	241	43
52142	26	254	32	193	32	30
52148	197	17	1	0	14	0
52154	96	105	205	181	3	193
52160	16	215	62	16	215	62
52166	7	215	62	8	215	62
52172	8	215	62	128	215	201
52178	62	8	184	32	221	62
52184	8	215	62	8	215	62
52190	128	215	195	10	202	62
58880	0	0	0	0	0	0
58886	0	0	0	1	2	4
58892	8	17	33	33	127	128
58898	0	0	0	131	131	131
58904	0	192	32	16	8	8
58910	4	4	25	37	37	34
58916	34	35	34	34	128	64
58922	64	64	64	64	163	188
58928	0	0	0	0	0	0
58934	128	64	0	0	0	0
58940	0	0	0	0	0	0
58946	0	0	0	0	0	0
58952	33	33	33	33	33	33
58958	33	33	131	131	255	255
58964	131	131	131	131	4	2
58970	2	2	4	4	8	8
58976	34	34	33	35	36	34
58982	17	8	36	196	6	4
58988	252	131	128	128	32	32
58994	160	65	66	131	15	10
59000	0	0	0	254	1	241
59006	243	1	8	20	20	36
59012	34	34	162	162	32	16
59018	16	16	8	8	4	4
59024	0	0	31	32	71	154
59030	98	36	0	0	248	70
59036	129	98	28	224	8	15
59042	136	144	208	127	63	31
59048	64	255	0	0	7	255
59054	255	192	0	144	191	252
59060	248	240	192	0	254	96
59066	255	1	1	2	2	2
59072	34	226	209	81	81	73
59078	73	141	2	1	1	0
59084	0	0	0	0	63	0
59090	0	128	96	16	12	3
59096	0	0	0	0	0	0
59102	0	0	0	0	0	0
59108	1	2	4	8	0	0
59114	0	31	224	0	1	2
59120	0	0	0	252	66	129
59126	0	0	2	2	4	4
59132	4	2	130	66	149	149
59138	245	149	149	141	137	137
59144	0	0	0	0	0	0
59150	0	0	0	0	0	0
59156	0	0	0	0	192	48
59162	14	1	0	0	0	0
59168	16	39	56	96	192	128
59174	128	143	61	200	16	32
59180	64	64	128	254	128	64
59186	48	12	2	1	1	2
59192	34	33	61	36	71	128
59198	0	0	145	145	161	161
59204	193	65	65	65	0	0
59210	0	0	0	0	0	0
59216	0	0	0	0	0	0
59222	0	0	0	0	0	0
59228	0	0	0	0	242	130
59234	132	132	136	72	40	31
59240	1	0	0	0	0	1
59246	6	248	4	136	72	48
59252	32	192	0	0	0	0
59258	0	0	0	0	0	0
59264	66	66	34	18	28	0
59270	0	0	0	0	0	12
59276	2	1	15	31	0	64
59282	128	152	160	192	248	236
59288	63	63	63	63	31	31
59294	15	7	246	242	242	246
59300	238	252	252	248	0	0
59306	0	14	1	0	48	13
59312	124	236	108	124	60	220
59318	196	160	3	99	30	4
59324	250	121	56	24	144	16
59330	128	64	64	0	0	0
59336	0	32	112	120	63	28
59342	15	15	0	0	0	0
59348	192	96	176	248	15	15
59354	7	3	1	0	0	0
59360	252	254	254	252	248	240
59366	96	0	14	25	32	32
59372	67	100	36	56	0	128

```
59378 128 254 2   129 121 5
59384 36  68  68  68  67  134
59390 121 0   5   170 202 9
59396 197 93  38  248 0   120
59402 76  70  67  32  44  74
59408 14  50  66  66  68  136
59414 80  176 73  70  47  18
59420 4   4   2   1   88  72
59426 68  162 122 42  36  192
59432 97  146 142 129 128 79
59438 63  7   130 74  122 143
59444 10  202 242 192 0   0
59450 56  124 191 1   0   0
59456 31  124 248 243 254 250
59462 224 0   3   14  31  63
59468 61  95  127 125 192 248
59474 252 118 254 255 255 255
59480 127 127 119 126 127 47
59486 31  0   239 255 255 255
59492 222 254 248 192 3   3
59498 3   3   3   3   207 255
59504 192 192 192 192 192 192
59510 204 255 1   1   7   15
59516 15  63  127 255 128 192
59522 192 224 240 252 254 254
59528 255 255 255 255 255 255
59534 255 255 254 252 251 243
59540 247 207 191 127 127 191
59546 223 207 247 243 253 254
59552 0   0   0   0   0   129
59558 199 255 0   0   60  126
59564 255 255 255 255 255 254
59570 253 251 251 199 63  255
59576 255 31  239 247 247 249
59582 254 255 24  56  89  123
59588 255 255 255 255 0   128
59594 192 192 192 160 184 191
59600 255 255 255 255 126 121
59606 135 255 255 131 125 126
59612 255 255 255 255 0   0
59618 0   0   0   0   0   0
59624 0   0   0   0   0   0
59630 0   0   0   0   0   0
59636 0   0   0   0   0   0
59642 0   219 219 0   0   0
59648 0   0   0   0   0   0
59654 0   0   0   0   0   0
59660 0   0   0   0   3   2
59666 4   4   8   8   16  16
59672 0   248 22  17  16  48
59678 56  40  0   0   0   192
59684 56  7   192 160 0   0
59690 0   0   0   0   192 32
59696 0   0   0   0   0   0
59702 0   0   0   0   0   0
59708 0   0   0   0   0   0
59714 0   0   0   0   0   0
59720 0   0   0   0   7   4
59726 28  23  16  16  32  32
59732 192 64  64  192 42  75
59738 83  83  145 161 162 162
59744 96  144 72  92  71  57
59750 1   1   16  8   4   130
59756 67  71  72  72  0   0
59762 0   1   3   131 97  28
59768 0   0   0   42  170 255
59774 42  42  0   0   0   128
59780 192 248 192 128 20  36
59786 47  40  104 239 104 40
59792 128 128 129 129 129 129
59798 129 129 162 162 34  66
59804 66  66  65  64  1   127
59810 168 168 169 121 8   144
59816 80  80  160 32  160 192
59822 192 192 6   5   9   18
59828 20  36  40  40  32  240
59834 120 44  7   5   9   10
59840 0   0   0   0   0   192
59846 64  32  47  40  40  39
59852 20  28  7   2   129 128
59858 128 128 128 128 128 143
59864 39  175 160 255 0   0
59870 0   255 249 255 0   255
59876 0   0   0   255 128 128
59882 128 128 0   0   0   252
59888 72  80  144 160 160 160
59894 160 144 18  20  37  37
59900 37  37  20  18  160 96
59906 176 200 246 252 240 224
59912 3   1   1   1   1   1
59918 0   0   136 4   2   2
59924 2   254 1   1   0   0
59930 0   0   0   0   0   0
59936 0   0   0   0   0   1
59942 2   4   8   16  32  64
59948 128 255 0   0   80  72
59954 40  41  30  224 0   0
59960 10  10  15  240 0   0
59966 0   0   64  64  128 0
59972 0   0   0   0   0   0
59978 0   0   0   0   0   0
59984 0   0   0   0   0   0
59990 0   0   0   128 255 0
59996 0   0   0   0   8   16
60002 224 0   0   0   0   0
60008 0   0   0   0   0   0
60014 0   0   0   0   0   0
60020 0   0   0   0   0   0
60026 0   0   0   0   0   0
60032 0   0   0   0   0   0
60038 0   0   7   31  61  121
60044 121 127 63  28  224 248
60050 188 158 158 254 252 56
60056 11  19  36  66  33  16
60062 8   0   208 200 36  66
60068 132 8   16  0   7   15
60074 29  57  57  59  31  29
60080 224 240 184 156 156 220
60086 248 184 13  13  15  15
60092 10  10  10  7   176 176
60098 240 240 80  80  80  224
60104 7   24  32  94  82  158
60110 142 128 224 24  4   122
60116 74  121 113 1   128 135
60122 138 154 175 184 128 255
60128 1   225 81  89  245 29
60134 1   255 14  25  32  32
60140 67  100 36  56  0   128
60146 128 254 2   129 121 5
60152 36  68  68  68  67  134
60158 121 0   5   170 202 9
60164 197 93  38  248 0   120
60170 76  70  67  32  44  74
60176 14  50  66  66  68  136
60182 80  176 73  70  47  18
60188 4   4   2   1   88  72
60194 68  162 122 42  36  192
60200 195 252 124 62  31  5
60206 7   0   192 32  16  60
60212 255 80  240 192 192 32
60218 16  60  255 80  240 192
60224 255 248 192 0   128 240
60230 255 255 255 63  3   0
60236 1   7   255 255 255 255
60242 252 192 0   224 252 255
60248 255 129 0   0   0   0
60254 7   255 255 127 15  1
60260 7   63  255 255 255 255
60266 195 0   227 255 255 255
60272 3   7   14  15  239 255
60278 247 95  128 192 224 96
60284 224 176 240 240 126 63
60290 23  31  14  15  15  15
60296 240 240 243 167 237 255
60302 254 190 15  15  13  15
60308 15  15  11  15  236 248
60314 240 96  224 224 224 224
60320 0   240 243 167 237 255
60326 254 190 15  15  13  15
60332 15  15  11  15  236 248
60338 240 96  224 224 224 224
60344 0   0   0   0   0   0
60350 0   0   0   0   0   0
60356 0   0   0   0   0   0
60362 0   0   0   0   0   0
60368 0   0   0   0   0   0
60374 0   0   0   0   0   0
60380 0   0   0   0   0   0
60386 0   0   0   0   0   0
60392 0   0   0   0   0   0
60398 0   0   0   0   0   0
60404 0   0   0   0   0   128
60410 192 255 255 192 128 0
60416 0   0   0   0   0   0
60422 0   0   240 254 255 127
60428 127 127 63  63  0   0
60434 224 252 255 255 255 255
60440 0   3   29  34  196 132
60446 196 195 0   254 65  160
60452 160 160 176 200 0   0
60458 128 64  32  32  16  8
60464 0   0   0   0   0   0
60470 0   0   0   0   0   0
60476 0   0   0   0   0   0
60482 0   0   0   0   0   0
60488 63  31  31  31  31  15
60494 15  15  255 255 255 255
60500 255 255 255 255 229 228
60506 248 248 252 252 254 254
60512 55  200 9   12  9   63
60518 65  38  8   136 143 144
60524 224 128 0   0   0   0
60530 0   225 253 255 255 255
60536 0   0   0   255 151 251
60542 255 255 0   0   0   128
60548 248 128 128 128 7   7
60554 7   7   3   3   3   1
60560 255 255 255 255 255 255
60566 255 255 254 255 255 255
60572 255 255 255 255 47  63
60578 127 255 255 255 255 255
60584 255 255 255 255 255 255
60590 255 255 255 255 255 255
60596 255 255 255 255 255 255
60602 255 255 255 255 255 255
60608 224 248 254 255 254 252
60614 248 240 1   1   0   0
60620 0   0   0   0   255 255
60626 0   255 127 63  31  15
60632 255 255 0   255 255 255
```

```
60638 255 255 255 255 0   255
60644 255 255 255 255 255 255
60650 0   255 255 255 255 255
60656 255 255 0   255 255 255
60662 255 254 255 255 0   254
60668 248 224 128 0   192 128
60674 0   0   0   0   0   0
60680 0   0   0   0   0   0
60686 0   0   7   3   1   0
60692 0   0   0   0   255 248
60698 243 240 113 56  28  15
60704 255 0   255 32  254 0
60710 0   3   255 127 63  62
60716 120 224 128 0   248 224
60722 128 0   0   0   0   0
60728 0   0   0   0   0   0
60734 0   0   0   0   0   0
60740 0   0   0   0   0   0
60746 0   0   0   0   0   0
60752 0   0   0   0   0   0
60758 0   0   7   3   1   0
60764 0   0   0   0   252 240
60770 192 0   0   0   0   0
60776 0   0   0   0   0   0
60782 0   0   0   0   0   0
60788 0   0   0   0   0   0
60794 0   0   0   0   0   0
60800 0   0   0   0   0   0
60806 0   0   15  63  127 120
60812 240 224 224 224 128 192
60818 192 64  0   126 124 0
60824 224 224 224 240 120 127
60830 63  15  0   124 126 0
60836 64  192 192 128 195 199
60842 206 236 252 252 124 126
60848 207 207 239 111 111 126
60854 126 126 63  31  7   0
60860 66  246 102 76  254 252
60866 240 0   130 207 198 98
60872 11  11  27  27  27  27
60878 27  27  160 160 176 176
60884 176 176 176 176 27  59
60890 59  243 243 227 227 195
60896 176 184 184 190 158 158
60902 142 134 14  25  32  32
60908 67  100 36  56  0   128
60914 128 254 2   129 121 5
60920 36  68  68  68  67  134
60926 121 0   5   170 202 9
60932 197 93  38  248 0   120
60938 76  70  67  32  44  74
60944 14  50  66  66  68  136
60950 80  176 73  70  47  18
60956 4   4   2   1   88  72
60962 68  162 122 42  36  192
60968 227 252 126 127 63  31
60974 12  7   192 32  28  63
60980 254 248 48  224 0   127
60986 64  95  240 23  20  20
60992 0   248 9   255 1   255
60998 2   2   63  48  255 1
61004 1   255 1   7   252 5
61010 5   5   63  32  254 130
61016 0   252 7   4   196 127
61022 64  64  0   0   248 8
61028 255 128 191 161 0   0
61034 0   0   0   0   0   0
61040 133 253 1   255 16  31
61046 0   0   2   2   254 35
61052 32  224 63  0   65  127

61058 4   252 160 160 160 63
61064 224 128 255 128 128 255
61070 12  252 64  64  255 128
61076 255 144 31  0   40  40
61082 232 15  250 2   254 0
61088 129 66  36  24  24  36
61094 66  129 255 0   0   0
61100 0   0   0   255 0   1
61106 6   24  32  64  76  48
61112 0   240 13  2   4   8
61118 0   0   0   0   128 96
61124 16  8   8   8   32  64
61130 64  64  64  32  16  15
61136 0   0   0   128 129 64
61142 32  223 8   8   16  16
61148 32  224 64  128 0   0
61154 0   0   0   0   0   0
61160 0   0   0   0   0   0
61166 0   0   0   0   0   0
61172 0   0   0   0   0   139
61178 32  255 252 0   65  8
61184 0   0   0   0   0   0
61190 0   0   0   0   0   0
61196 0   1   1   3   0   0
61202 0   24  239 239 239 239
61208 0   0   0   0   192 224
61214 176 176 0   0   0   0
61220 0   0   89  151 0   0
61226 0   0   0   0   0   0
61232 0   0   0   0   0   0
61238 0   0   0   0   0   0
61244 0   0   0   0   0   0
61250 0   0   0   0   0   0
61256 3   7   8   30  31  31
61262 31  31  247 250 250 251
61268 11  243 216 219 116 248
61274 252 252 254 255 255 255
61280 20  37  37  26  42  69
61286 192 192 128 0   12  244
61292 74  138 146 147 0   0
61298 0   0   0   0   0   192
61304 0   0   0   0   0   0
61310 0   0   0   0   0   0
61316 0   0   0   0   15  15
61322 31  30  49  31  31  31
61328 191 223 239 79  187 252
61334 255 199 255 199 59  253
61340 253 125 141 225 227 229
61346 225 255 253 253 253 251
61352 231 15  63  255 255 127
61358 255 122 249 223 159 155
61364 219 91  219 211 96  176
61370 248 254 255 247 108 219
61376 0   0   0   0   0   128
61382 192 192 30  14  14  14
61388 15  7   0   0   247 251
61394 252 255 255 255 255 3
61400 254 224 111 111 113 56
61406 38  67  125 124 127 63
61412 223 24  255 63  239 112
61418 7   216 43  246 239 255
61424 150 117 133 62  227 221
61430 63  247 246 196 189 58
61436 218 132 243 251 192 128
61442 128 0   128 0   0   0
61448 0   0   0   0   0   0
61454 0   0   0   0   0   0
61460 0   0   0   0   7   7
61466 7   15  15  7   3   0
61472 162 189 251 195 247 247
61478 249 253 63  199 251 253

61484 63  159 227 190 255 188
61490 224 128 128 0   0   0
61496 252 112 0   0   0   0
61502 0   0   0   0   0   0
61508 0   0   0   0   0   0
61514 0   0   0   0   0   0
61520 0   0   0   0   0   0
61526 0   0   0   0   0   0
61532 0   0   0   0   31  3
61538 1   1   1   0   0   0
61544 220 232 224 224 192 128
61550 0   0   0   0   0   0
61556 0   0   0   0   0   0
61562 0   0   0   0   0   0
61568 0   0   0   0   0   0
61574 0   0   129 193 178 140
61580 160 88  32  33  128 64
61586 64  32  112 200 218 253
61592 35  35  33  32  20  19
61598 8   8   41  105 169 201
61604 6   248 240 32  16  48
61610 208 144 145 176 208 16
61616 64  32  24  4   2   194
61622 60  32  16  32  64  129
61628 129 130 68  56  160 144
61634 136 4   2   130 66  60
61640 112 80  80  81  81  81
61646 112 16  32  32  16  16
61652 16  16  144 96  16  33
61658 33  34  66  66  69  56
61664 32  16  16  16  16  16
61670 16  224 16  18  18  18
61676 18  18  17  16  32  32
61682 32  32  32  32  32  192
61688 16  32  36  36  66  66
61694 69  56  64  32  32  16
65368 126 62  6   2   0   0
65374 0   0   129 129 241 143
65380 129 129 113 15  65  65
65386 64  64  64  64  64  255
65392 157 149 213 247 127 62
65398 0   255 194 194 130 130
65404 2   2   2   255 129 129
65410 143 241 129 129 142 240
65416 0   0   0   0   2   6
65422 56  194 15  113 129 129
65428 143 241 129 129 255 64
65434 71  69  71  68  68  64
65440 255 0   119 84  87  81
65446 119 0   255 2   114 34
65452 34  34  34  2   240 142
65458 129 129 241 143 129 129
65464 1   2   4   8   16  48
65470 120 252 5   6   225 64
65476 225 255 255 0   201 9
65482 49  193 225 241 243 7
65488 248 240 240 224 224 192
65494 192 128 1   2   4   8
65500 17  38  72  156 255 1
65506 249 253 61  29  157 29
65512 255 255 254 254 252 252
65518 248 248 60  66  66  66
65524 90  90  90  255 56  12
65530 56  108 108 56  96  56
```

Okay, die Arbeit ist überstanden, und nun kann das Spiel beginnen!

ZX-Spectrum:

Ausschnittkopierer

Mit dieser MC-Routine ist es mit dem 16/48 KByte Spectrum möglich, einen vom Benutzer begrenzten Teilbereich eines Basicprogramms abzuspeichern ohne das Programm zu verändern.

Jeder, der auf dem Spectrum schon einmal mit Pascal programmiert hat, wird den Vorteil zu schätzen wissen, daß es im Hisoft Pascal möglich ist, nur einen bestimmten Programmbereich abzuspeichern, anstatt immer das ganze Programm sichern zu müssen. Dieses ist vor allem dann von Vorteil, wenn man z. B. längere mathematische Formeln, die bereits in einem bestehenden Programm verwendet wurden, in ein neues Programm einbauen will. Vorher mußten die Zeilen, die die Formel einschließen gelöscht oder die Formeln selbst abgetippt werden. Diesem Zustand schafft die folgende MC-Routine Abhilfe. Dieses ermöglicht es, vom Benutzer begrenzte Programmblöcke zu sichern ohne das gesamte Basicprogramm zu verändern.

Nach Eingabe des Basicprogramms müssen Sie sich eine Startadresse überlegen und schützen diesen Bereich durch Eingabe von CLEAR Startadresse-1 (z. B. CLEAR 64999). Anschließend geben Sie RUN ein, und das Programm fragt Sie nach der Startadresse, ab der das Programm in den Speicher gepoket werden soll. Am Ende jeder Dataliste befindet sich eine Checksumme. Haben Sie Eingabefehler begangen, so wird die entsprechende DATA-Zeile angezeigt, worauf Sie den Fehler beheben und das Programm neu starten müssen. Andernfalls wird jede Zeile mit einem O.K. abgeschlossen und es erscheint die Meldung „Start the tape . . .“, worauf Code und Basicprogramm abgespeichert werden können.

Der Code kann nun zu dem Programm, aus dem man einen Ausschnitt abspeichern möchte, hinzugeladen werden, nachdem man den entsprechenden Bereich durch ein CLEAR geschützt hat. Er wird in folgender Weise aufgerufen:

RANDOMIZE USR Startadresse: DATA erste abzuspeichernde Zeile, letzte abzuspeichernde Zeile, worauf am oberen Bildschirmrand „Programmausschnittkopierer“ und „Start the tape . . . “ erscheint und der gewünschte Bereich auf Band unter dem Namen „PARTCOPY“ gesichert werden kann.

Beispiel:

RANDOMIZE USR 65000: DATA 30,100 würde den Bereich zwischen Zeile 30 und 100 (einschließlich Zeile 30 und 100) abspeichern. Existiert eine der abgegebenen Zeilennummern nicht, so wählt das Programm die nächstgrößere aus. Wird eine Zeilennummer angegeben, die größer als die letzte existierende ist, so erscheint Fehlermeldung 8 End of file.

```
   1 REM * Ausschnittkopierer *
   2 REM **** Version  1.0 ****
   3 REM *** Copyright 1985 ***
   4 REM ** by  Oliver Thimm **
   5 REM ** 3000 Hannover 91 **
  10 DEF FN a(x)=INT (x/256)
  20 DEF FN b(x)=(x-256*FN a(x))
  30 CLS : INPUT "Startadresse ?
 ";start
  40 LET x=start+96
  50 LET a=FN b(x): LET b=FN a(x
)
  60 LET x=x+2
  70 LET c=FN b(x): LET d=FN a(x
)
  80 LET x=x+17
  90 LET e=FN b(x): LET f=FN a(x
)
 100 LET x=start: LET sum=0
 110 FOR n=500 TO 590 STEP 10: R
ESTORE n
 120 FOR m=0 TO 14
 130 READ byte: POKE x+m,byte: L
ET sum=sum+byte
 140 NEXT m
 150 READ tot: IF tot<>sum THEN
BEEP .5,10: PRINT "Error in Line
 ";n: STOP
 160 PRINT "Line ";n;" O.K"
 170 LET sum=0: LET x=x+15: NEXT
 n
 180 SAVE "TeilesaveCODE "CODE s
tart,150: SAVE "Teilesaver"
 190 CLS : PRINT "Band fuer 'VER
IFY' zurueckspulen"
 200 VERIFY ""CODE : VERIFY ""
 210 CLS : PRINT "Solle z.B. die
 Zeilen 100-210   dieses Program
ms gesaved werden,so geben Sie f
olgendes ein:": PRINT : PRINT "R
ANDOMIZE USR ";start;":DATA 100,
210": STOP
 500 DATA 231,231,205,251,36,205
,162,45,197,231,231,205,251,36,2
05,2722
 510 DATA 162,45,209,197,213,225
,205,110,25,193,229,197,225,205,
110,2550
 520 DATA 25,35,35,78,35,70,35,3
5,237,74,229,237,75,75,92,1367
```

```
530 DATA 43,43,237,66,242,a,b,2
25,209,213,237,82,229,205,107,21
38+a+b
540 DATA 13,62,2,205,1,22,17,e,
f,1,29,0,205,60,32,649+e+f
550 DATA 209,225,221,33,c,d,221
,115,11,221,114,12,221,115,15,17
33+c+d
560 DATA 221,114,16,195,112,9,2
07,7,0,80,65,82,84,67,79,1338
570 DATA 80,89,32,32,218,3,0,12
8,218,3,32,32,32,80,82,1061
580 DATA 79,71,82,65,77,77,65,8
5,83,83,67,72,78,73,84,1141
590 DATA 84,75,79,80,73,69,82,6
9,82,0,0,0,0,0,0,693
```

PAC-BOY für den Apple II+/IIE/IIC

Versuchen Sie, alle Punkte aufzufressen. Aber Vorsicht: Das PAC-Monster versucht, Sie daran zu hindern. Keine Angst! Sie können die Super PAC-Pille auffressen, dann besitzen Sie für einige Zeit die Kraft, das Monster zu versteinern.

Steuerung:
I = OBEN J = LINKS K = RECHTS
M = UNTEN

Eintippen:
Tippen Sie erst Listing 1 ein, und speichern Sie es auf Diskette ab. Speichern Sie das zweite Programm mit SAVE PAC-BOY ab.

„Unsere Abbildung zeigt PAC-BOY nach erfolgreich überstandenem Kampf gegen das grausame Monster. Die Strapazen sind ihm ja noch deutlich anzusehen, aber allem Anschein nach sind PAC-BOY auch die Super-Pac-Pillen ein wenig auf den Magen geschlagen! Aber das weiß ja jeder: Allzuviel ist ungesund!"



Listing – Teil eins

```
1   HGR2 : HCOLOR= 3
2   HPLOT 40,30 TO 40,10 TO 50,10 TO
      50,20 TO 40,20
3   HPLOT 60,30 TO 60,10 TO 70,10 TO
      70,30: HPLOT 60,20 TO 70,20
4   HPLOT 90,10 TO 80,10 TO 80,30 TO
      90,30: HPLOT 100,20 TO 110,2
      0
5   HPLOT 120,10 TO 130,10 TO 130,
      30 TO 120,30 TO 120,10: HPLOT
      120,20 TO 130,20
6   HPLOT 140,10 TO 150,10 TO 150,
      30 TO 140,30 TO 140,10
7   HPLOT 160,10 TO 160,20 TO 170,
      20 TO 170,10: HPLOT 165,20 TO
      165,30
8   HPLOT 60,40 TO 65,40 TO 65,50 TO
      60,50 TO 60,40: HPLOT 60,45 TO
      65,45
9   HPLOT 70,45 TO 70,50 TO 75,50 TO
      75,45: HPLOT 72,50 TO 72,55
10   HPLOT 90,50 TO 90,60 TO 95,60
       TO 95,50 TO 90,50: HPLOT 95
       ,55 TO 90,55
11  HPLOT 100,60 TO 100,50 TO 105
      ,50 TO 105,60: HPLOT 100,55 TO
      105,55
12  HPLOT 110,60 TO 110,50 TO 115
      ,60 TO 115,50
13  HPLOT 120,60 TO 120,50 TO 125
      ,50 TO 125,60: HPLOT 120,55 TO
      125,55
14  HPLOT 130,60 TO 130,50 TO 135
      ,60 TO 135,50
15  HPLOT 140,60 TO 140,50 TO 145
      ,50 TO 145,60: HPLOT 140,55 TO
      145,55
```





```
16  HPLOT 70,70 TO 80,70 TO 80,80
     TO 70,80 TO 70,70: HPLOT 77
    ,72 TO 73,72 TO 73,77 TO 77,
    77
17  HPLOT 90,70 TO 90,80: HPLOT 9
    5,70 TO 100,70 TO 100,75 TO
    95,75 TO 95,70: HPLOT 100,75
     TO 100,80
18  HPLOT 105,70 TO 110,70 TO 110
    ,80 TO 105,80 TO 105,70: HPLOT
    105,75 TO 110,75
19  HPLOT 120,70 TO 115,70 TO 115
    ,75 TO 120,75 TO 120,80 TO 1
    15,80
20  HPLOT 105,100 TO 110,110 TO 1
    20,115 TO 110,120 TO 105,130
     TO 100,120 TO 90,115 TO 100
    ,110 TO 105,100: HPLOT 105,1
    00 TO 105,130: HPLOT 120,115
     TO 90,115
21  HPLOT 105,85 TO 105,95: HPLOT
    130,110 TO 125,115 TO 130,12
    0: HPLOT 125,110 TO 125,120
22  HPLOT 100,140 TO 100,130 TO 1
    05,135 TO 110,130 TO 110,140

23  HPLOT 85,110 TO 85,120 TO 80,
    120 TO 80,115
24  HPLOT 50,160 TO 50,180: HPLOT
    60,160 TO 60,180: HPLOT 60,1
    70 TO 50,170: HPLOT 70,160 TO
    70,180
25  HPLOT 80,160 TO 90,160: HPLOT
    85,160 TO 85,180
26  HPLOT 110,180 TO 110,160 TO 1
    20,160 TO 120,180: HPLOT 120
    ,170 TO 110,170
27  HPLOT 130,180 TO 130,160 TO 1
    40,180 TO 140,160: HPLOT 150
    ,160 TO 150,170 TO 160,170 TO
    160,160: HPLOT 155,170 TO 15
    5,180
28  HPLOT 190,160 TO 180,170 TO 1
    90,180: HPLOT 180,160 TO 180
    ,180: HPLOT 210,160 TO 200,1
    60 TO 200,180 TO 210,180: HPLOT
    200,170 TO 210,170
29  HPLOT 220,160 TO 220,170 TO 2
    30,170 TO 230,160: HPLOT 225
    ,170 TO 225,180
30  HPLOT 40,150 TO 240,150 TO 24
    0,190 TO 40,190 TO 40,150
31  GET ZZ$: PRINT ZZ$
32  PRINT  CHR$ (4)"RUN PAC-BOY"
```

PAC MAN – Teil zwei ➡

```
0 S = 0:L = 3: DIM VA(14,11)
1  HOME : NORMAL : NOTRACE
5  GOSUB 6000: GOSUB 10000
10  FOR Y = 0 TO 11
15  FOR X = 0 TO 14
20  READ A:VA(X,Y) = A
22  IF S > 2230 AND A / 13 =  INT
     (A / 13) THEN A = (A / 13) *
      SGN (S - 4470):VA(X,Y) = A:
     NE = 1
25  NEXT X
27  NEXT Y
28  IF S > 0 THEN VA(0,5) = 0:VA(
     1,5) = 30:VA(14,5) = 0:VA(13
     ,5) = 70:TB = 1
30  GOSUB 5000: ROT= 0: XDRAW 3 AT
     129,145: XDRAW 2 AT 139,145:
      XDRAW 3 AT 149,145
32  ROT= 0: SCALE= 1: FOR P = 1 TO
     L: XDRAW 2 AT P * 10 + 119,1
     0: NEXT P
35 V = 10:DI = 0:D =  - 1:A = 139
     :OA = A:B = 64:OB = B: ROT=
     0: XDRAW 3 AT OA,OB
40 X = 139:Y = 114:R = 0
45 OX = X:OY = Y:RO = R: ROT= RO:
      XDRAW 2 AT OX,OY
47  HOME : IF NE = 1 THEN  INVERSE
     : VTAB (21): PRINT "    NO T
     UNNEL AND NO ENERGIZERS
       ": NORMAL : GOTO 50
48  HOME : IF TB = 1 THEN  INVERSE
     : VTAB (21): PRINT "      THE
      TUNNEL IS BLOCKED
       ": NORMAL
50  VTAB (22): FLASH : PRINT "
        HIT ANY KEY TO START
             ": POKE  - 16368,0: WAIT
      - 16384,128: NORMAL
60  HOME : VTAB (21): PRINT "
          SCORE="
100  FOR T = 1 TO 2 STEP 0
110 K =  PEEK ( - 16384)
120 XA = (OX - 69) / 10:YA = (OY -
     24) / 10:PA = VA(XA,YA)
125  IF K = 201 THEN R = 0: IF PA
      / 2 =  INT (PA / 2) THEN Y =
     Y - V
130  IF K = 205 THEN R = 32: IF P
     A / 5 =  INT (PA / 5) THEN Y
      = Y + V
135  IF K = 202 THEN R = 48: IF P
     A / 7 =  INT (PA / 7) THEN X
      = X - V: IF X < 69 THEN X =
     209
140  IF K = 203 THEN R = 16: IF P
     A / 3 =  INT (PA / 3) THEN X
```

```
     = X + V: IF X > 209 THEN X =
    69
142 XA = (X - 69) / 10:YA = (Y -
    24) / 10:PA = VA(XA,YA)
145  ROT= RO: XDRAW 2 AT OX,OY: ROT=
    R: XDRAW 2 AT X,Y
146 OX = X:OY = Y:RO = R
147  IF (A = X) * (B = Y) * (SF =
    0) THEN  GOTO 1000
150  IF PA > 0 AND PA / 13 <  >  INT
    (PA / 13) THEN  ROT= 0: XDRAW
    1 AT X,Y: POKE 768,96: POKE
    769,3: CALL 770:VA(XA,YA) =
     - PA:DT = DT - 1:S = S + 10
    : VTAB (21): HTAB (20): PRINT
    S: IF DT = 0 THEN  GOTO 2000

152  IF SF = 1 THEN  ROT= 0: RETURN

155  IF PA > 0 AND PA / 13 =  INT
    (PA / 13) THEN  ROT= 0: XDRAW
    4 AT X,Y: FOR P = 10 TO 1 STEP
     - 1: POKE 768,P: POKE 769,3
    : CALL 770: NEXT P:VA(XA,YA)
     =  - PA: GOSUB 600
200 GH = VA((OA - 69) / 10,(OB -
    24) / 10)
205 D =  - D:DI = 0
207  IF (Y = B) THEN  GOTO 215
210  IF D < 0 OR (X = A) THEN  GOTO
    255
215  IF (X > A) THEN  GOTO 240
220  IF GH / 7 =  INT (GH / 7) THEN
    A = A - V: GOTO 320
230  GOTO 250
240  IF GH / 3 =  INT (GH / 3) THEN
    A = A + V: GOTO 320
250 DI = DI + 1:D =  - 1: IF DI =
    2 THEN  GOTO 285
255  IF Y > B THEN  GOTO 270
260  IF GH / 2 =  INT (GH / 2) THEN
    B = B - V: GOTO 320
265  GOTO 275
270  IF GH / 5 =  INT (GH / 5) THEN
    B = B + V: GOTO 320
275 DI = DI + 1:D = 1: IF DI = 2 THEN
     GOTO 285
280  GOTO 215
285  IF (X - A) *  SGN (X - A) >
    (Y - B) *  SGN (Y - B) THEN
     GOTO 300
290  IF GH / 7 =  INT (GH / 7) THEN
    A = A - V:D = 1: GOTO 320
295 A = A + V:D = 1: GOTO 320
300  IF GH / 2 =  INT (GH / 2) THEN
    B = B - V:D =  - 1: GOTO 320
305 B = B + V:D =  - 1
320  IF A > 209 THEN A = 69
323  IF A < 69 THEN A = 209
325  ROT= 0: XDRAW 3 AT OA,OB: XDRAW
    3 AT A,B: IF (A = X) * (B =
    Y) THEN  GOTO 1000
330 OA = A:OB = B
400  NEXT T
600 SF = 1: ROT= 0: XDRAW 3 AT 12
    9,145: XDRAW 2 AT 139,145: XDRAW
    3 AT 149,145: XDRAW 2 AT 129
    ,145: XDRAW 3 AT 139,145: XDRAW
    2 AT 149,145
610  FOR Z = 1 TO 15
620  GOSUB 110: IF (X = A) * (Y =
    B) THEN  GOSUB 800: GOTO 700

630 P =  INT ( RND (1) * 3)
635 GH = VA((OA - 69) / 10,(OB -
    24) / 10)
640  IF P = 2 AND GH / 5 =  INT (
    GH / 5) THEN B = B + V
650  IF P = 0 AND GH / 2 =  INT (
    GH / 2) THEN B = B - V
660  IF P = 3 AND GH / 7 =  INT (
    GH / 7) THEN A = A - V: IF A
     < 69 THEN A = 209
670  IF P = 1 AND GH / 3 =  INT (
    GH / 3) THEN A = A + V: IF A
     > 209 THEN A = 209
680  ROT= 0: XDRAW 3 AT OA,AB: XDRAW
    3 AT A,B:OA = A:OB = B
690  IF (X = A) * (Y = B) THEN  GOSUB
    800
700  NEXT Z
710 SF = 0: ROT= 0: XDRAW 2 AT 12
    9,145: XDRAW 3 AT 139,145: XDRAW
    2 AT 149,145: XDRAW 3 AT 129
    ,145: XDRAW 2 AT 139,145: XDRAW
    3 AT 149,145
715  FOR P = 1 TO 15:Z =  PEEK ( -
    16336): NEXT P
720  RETURN
800  XDRAW 3 AT A,B:A = 139:B = 6
    4:OA = A:OB = B: FOR P = 1 TO
    5: PRINT  CHR$ (7);: NEXT P:
    Z = 15: XDRAW 3 AT OA,OB:S =
    S + 100: VTAB (21): HTAB (20
    ): PRINT S: RETURN
1000  ROT= R: XDRAW 2 AT X,Y: SCALE=
    2: FOR R = 0 TO 128 STEP 8: ROT=
    R: XDRAW 2 AT X,Y:D =  PEEK
    ( - 16336): NEXT R: POKE  -
    16368,0: HOME : VTAB (21): HTAB
    (15): PRINT "SCORE= ";S
1005  XDRAW 2 AT X,Y: ROT= 0: SCALE=
```



```
    1: XDRAW 3 AT A,B: XDRAW 2 AT
    L * 10 + 119,10:L = L - 1: IF
    L > 0 THEN 35
1010  PRINT "DO YOU WANT ANOTHER
                GAME ?";: FOR P =
    1 TO 500: NEXT P: POKE  - 16
    368,0: GET Q$: IF Q$ = "Y" THEN
     CLEAR : DIM VA(14,11):L = 3
    : GOTO 1
1020  TEXT : HOME : NEW : HOME
1100  REM ***************
2000  HOME : VTAB (22): FLASH : PRINT
    "PERFECT SCORE.WELL DONE
            ": FOR P = 1 TO 15: POKE
    768,16 - P: POKE 769,6: CALL
    770: NEXT P: RESTORE : IF SF
     = 1 THEN  POP
2005 SF = 0: GOTO 1
5000  HGR
5002  FOR Y = 0 TO 11
5005  FOR X = 0 TO 14
5010 A = VA(X,Y)
5015 K = 69 + X * 10:V = 24 + Y *
    10
5020  HCOLOR= 7
5050  IF A / 2 <  > INT (A / 2) THEN
     HPLOT K - 5,V - 5 TO K + 5,
    V - 5
5055  IF A / 3 <  > INT (A / 3) THEN
     HPLOT K + 5,V - 5 TO K + 5,
    V + 5
5060  IF A / 5 <  > INT (A / 5) THEN
     HPLOT K - 5,V + 5 TO K + 5,
    V + 5
5065  IF A / 7 <  > INT (A / 7) THEN
     HPLOT K - 5,V - 5 TO K - 5,
    V + 5
5067  IF A > 0 AND A / 13 =  INT
    (A / 13) THEN  XDRAW 4 AT K,
    V: GOTO 5400
5070  IF A > 0 THEN  XDRAW 1 AT K
    ,V:DT = DT + 1
5400  NEXT X
5410  NEXT Y
5420  HPLOT 0,0 TO 64,19: HPLOT 2
    79,0 TO 214,19: HPLOT 0,159 TO
    64,139: HPLOT 279,159 TO 214
    ,139
5500  RETURN
6000  FOR X = 770 TO 792
6010  READ Y
6020  POKE X,Y
6030  NEXT X
6040  DATA 173,48,192,136,208,5,2
    06,1,3,240,9,202,208,245,174
    ,0,3,76,2,3,96,0,0
6050  RETURN
10000  HGR2 : POKE 232,16: POKE 2
    33,64
10005  TEXT
10010  SCALE= 1: ROT= 0
10020  FOR X = 16400 TO 16460
10030  READ Y
10040  POKE X,Y
10050  NEXT X
10055  RETURN
10060  DATA 4,0,10,0,16,0,31,0,51
    ,0
10070  DATA 42,36,63,54,21,0
10080  DATA 18,27,39,36,44,44,9,4
    9,53,54,62,62,63,39,0
10090  DATA 46,30,46,9,36,39,37,3
    9,39,63,46,62,35,23,46,21,63
    ,54,6,0
10100  DATA 53,30,60,60,12,37,21,
    21,6,0
20000  DATA 15,21,105,21,105,21,3
    5,0,15,21,105,21,105,21,35
20010  DATA 10,15,42,35,6,35,30,2
    73,70,15,14,15,42,35,10
20030  DATA 30,910,0,30,21,210,70
    ,0,30,210,21,70,0,390,70
20040  DATA 10,6,105,14,15,14,30,
    21,70,6,35,6,105,14,10
20050  DATA 6,35,60,21,210,21,-21
    0,-21,-210,21,210,21,70,15,1
    4
20060  DATA -21,210,210,21,210,21
    ,-70,0,-30,21,210,21,210,210
    ,-21
20070  DATA 15,14,30,35,6,35,-6,-
    105,-14,15,14,15,70,6,35
20080  DATA 30,105,14,30,21,210,1
    05,42,105,210,21,70,6,105,70

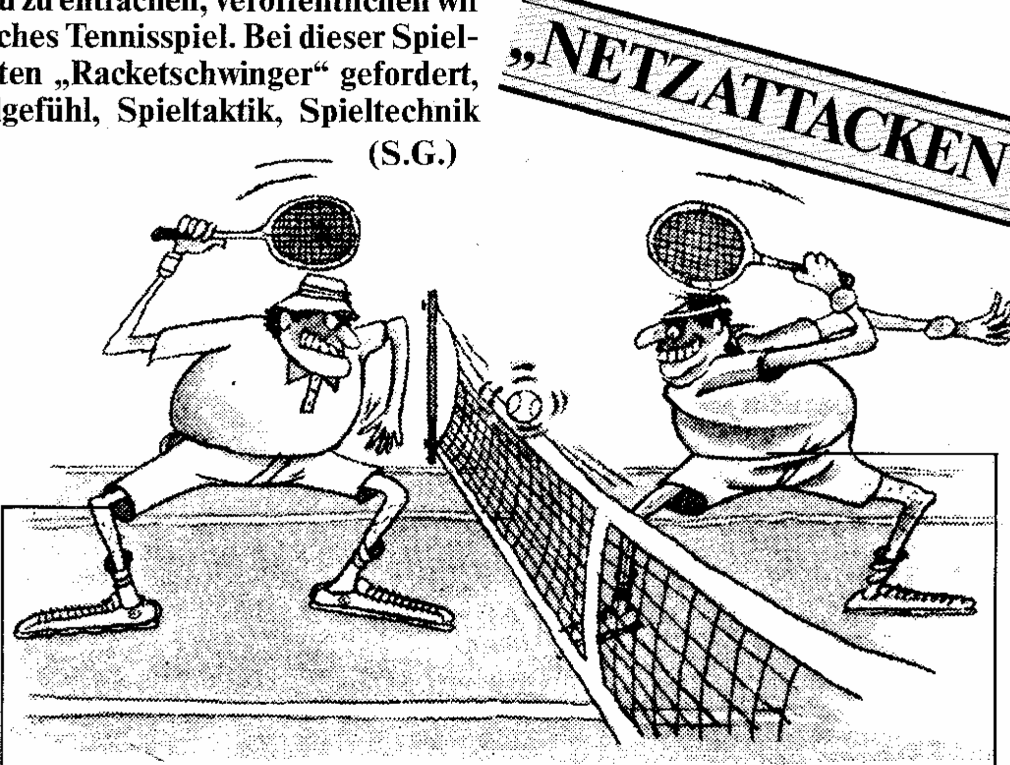
20090  DATA 10,30,21,70,15,14,30,
    105,70,6,35,30,21,70,10
20100  DATA 30,70,0,30,14,195,14,
    -10,6,455,6,70,0,30,70
20110  DATA 10,6,105,14,15,14,15,
    42,35,6,35,6,105,14,10
20120  DATA 6,21,42,21,42,21,14,0
    ,6,21,42,21,42,21,14
```

Auf geht's! PAC BOY wartet auf seine Bewährungsprobe. Keine Angst vor dem Monster!

Top-Tennis ... + Top-Tennis ... + Top-Tennis ... + Top-Tennis ... +

Eine packende Tennissimulation für Atari 600 (+64k) und 800 XL

Um das bundesweite Tennisfieber neu zu entfachen, veröffentlichen wir an dieser Stelle ein wirklich realistisches Tennisspiel. Bei dieser Spielversion werden selbst die erfahrensten „Racketschwinger" gefordert, was Schlagreportoire, Timing, Ballgefühl, Spieltaktik, Spieltechnik und Spielwitz betrifft. (S.G.)



Hier sehen Sie Verleger (links) und Chefredakteur der COMPUTRONIC, die sich entschlossen haben, die lästigen Gehaltsverhandlungen diesmal bei einer Partie TOP-TENNIS auszufechten.

Das Spiel bietet zahlreiche Variationsmöglichkeiten: Volley, Passierschlag, Lob ... etc. Um den Ball zu schlagen, muß man die Ballrichtung mit dem Joystick bestimmen und dann den Feuerknopf betätigen. Die Härte des Schlages hängt von der Ballhöhe ab. Wird mit dem Joystick keine Richtung angegeben, so spielt man den Ball in Form eines Lobs mit Topspin-Effekt. Aufschlag: Wenn man den Feuerknopf drückt, wirft der Spieler den Ball nach oben, anschließend wird mit dem Joystick die Richtung bestimmt und der Feuerknopf noch einmal betätigt. Es sind zwei Aufschlaggeschwindigkeiten möglich. Zum besseren Verständnis sollte man einfach Practice-Mode wählen.

Um zum Menue zurückzukehren, muß die Starttaste vor dem Aufschlag betätigt werden. Während der Ballwechsel ist eine Wartepause möglich, indem man die Spacetaste drückt. Wenn das Spiel fortgesetzt werden soll, muß diese Taste erneut gedrückt werden.

TOP-TENNIS:

Programmaufbau

Achtung: Top-Tennis ist nur auf Cassette lauffähig! Das Programm muß mit dem Befehl „CLOAD" gestartet werden! Das Hauptprogramm ist in Maschinensprache geschrieben und daher in drei Teile untergliedert: 1. und 2. Teil – Datazeilen 3. Teil - Basic-Programm Maschinencode - Zeile 301 – 304

TOP-TENNIS – TEIL EINS

```
1 REM COPYRIGHT BERNARD DALLE 1985
2 REM ******************************
3 RESTORE 4500:FOR X=0 TO 71:READ C:POKE
 22000+X,C:NEXT X
4 FOR X=0 TO 71:READ C:POKE 22100+X,C:NE
XT X
5 RESTORE
10 POKE 710,9*16:POKE 712,9*16:POKE 752,
1:? CHR$(125)
20 POSITION 15,3:? "TOP TENNIS":POSITION
 17,6:? "PART 1"
30 POSITION 11,12:? "PLEASE WAIT 80 sec.
"
275 FOR X=0 TO 20:READ FG:POKE 20945+X,F
G:NEXT X
276 FOR X=0 TO 244:READ FG:POKE 20477+X,
FG:NEXT X:POSITION 11,12:? "    PLEASE W
AIT      "
278 FOR X=0 TO 62:READ GF:POKE 1536+X,GF
:NEXT X
282 FOR X=0 TO 214:READ FG:POKE 20730+X,
FG:NEXT X
284 FOR X=0 TO 31:READ GF:POKE 1664+X,GF
:NEXT X
286 FOR X=0 TO 24:READ FG:POKE 300+X,FG:
NEXT X
290 FOR X=0 TO 282:READ GF:POKE 30000+X,
GF:NEXT X:FOR X=0 TO 854:READ FG:POKE 30
300+X,FG:NEXT X
291 FOR X=0 TO 44:READ BYTE:POKE 370+X,B
YTE:NEXT X
294 FOR X=0 TO 859:READ FG:POKE 22200+X,
FG:NEXT X
300 FOR X=0 TO 10:READ BYTE:POKE 22080+X
,BYTE:NEXT X
310 FOR X=0 TO 30:READ BYTE:POKE 23070+X
,BYTE:NEXT X
320 FOR X=0 TO 62:READ BYTE:POKE 23500+X
,BYTE:NEXT X
330 FOR X=0 TO 26:READ FG:POKE 23400+X,F
G:NEXT X
340 FOR X=0 TO 62:READ BYTE:POKE 23600+X
,BYTE:NEXT X
350 FOR X=0 TO 26:READ FG:POKE 23430+X,F
```

```
G:NEXT X
360 FOR X=0 TO 62:READ BYTE:POKE 23700+X
,BYTE:NEXT X
370 FOR X=0 TO 62:READ BYTE:POKE 23800+X
,BYTE:NEXT X
380 FOR X=0 TO 243:READ BYTE:POKE 24000+
X,BYTE:NEXT X
390 FOR X=0 TO 213:READ BYTE:POKE 24300+
X,BYTE:NEXT X
400 FOR X=0 TO 28:READ BYTE:POKE 24600+X
,BYTE:NEXT X
405 RESTORE 7010:FOR X=0 TO 69:READ FG:P
OKE 25000+X,FG:NEXT X
406 FOR X=0 TO 129:READ BYTE:POKE 25500+
X,BYTE:NEXT X
408 FOR X=0 TO 84:READ C:POKE 26000+X,C:
NEXT X
410 POSITION 11,6:? "NOW LOADING PART 2"
411 POSITION 5,12:? "PARTS 1,2,3  writte
n by B.DALLE"
420 POKE 764,33:A=USR(26053)
1190 DATA 201,13,240,7,201,14,240,3,76,1
3,80,96,9,169,255,145,204,200,145,204,96
1200 DATA 104,104,104,170,104,133,205,10
4,133,204,189,120,2,76,209,81,201,7,240,
45,201,6,240,41
1205 DATA 201,5,240,37,201,10
1210 DATA 240,124,201,11,240,120,201,9,2
40,116,160,0,185,0,6,145,204,152,201,8,2
40,4,200,76,39,80,189,80
1220 DATA 6,157,0,208,76,241,80,189,80,6
,201,200,176,246,24,189,82,6,105,1,157,8
2,6,201,2,240,20,201,3,240
1230 DATA 32,160,0,185,9,6,145,204,152,2
01,8,240,40,200,76,88,80,160,0,185,18,6,
145,204,152,201,8,240,24
1240 DATA 200,76,104,80,169,0,157,82,6,1
68,185,27,6,145,204,152,201,8,240,4,200,
76,124,80,24,189,80,6,105,4
1250 DATA 157,80,6,157,0,208,76,241,80,1
89,80,6,201,50,144,81,24,189,84,6,105,1,
157,84,6,201,2,240,20,201,3
1260 DATA 240,32,160,0,185,36,6,145,204,
152,201,8,240,40,200,76,179,80,160,0,185
,45,6,145,204,152,201,8,240
1270 DATA 24,200,76,195,80,169,0,157,84,
6,168,185,54,6,145,204,152,201,8,240,4,2
00,76,215,80,56
1280 DATA 189,80,6,233,4,157,80,6,157,0,
208,96
1300 DATA 28,28,62,62,28,28,20,54,0
1310 DATA 7,119,114,62,50,48,232,132,6
1320 DATA 7,119,114,62,50,48,48,40,48
1330 DATA 7,119,114,62,50,48,60,70,128
1340 DATA 224,238,78,124,76,12,63,65,192
1350 DATA 224,238,78,124,76,12,12,20,12
1360 DATA 224,238,78,124,76,12,60,98,1
1400 DATA 104,32,44,1,104,104,170,76,30,
90,104,133,205,104,133,204,189,120,2,201
,6,240,23,201,10,240
1410 DATA 19,201,14,240,15,201,13,240,92
,201,9,240,88,201,5,240,84,76,199,81
1420 DATA 189,72,6,221,245,80,240,70,189
,64,6,24,105,1,157,64,6,201,3,240
1430 DATA 20,201,4,240,16,160,0,185,128,
6,145,204,152,201,16,240,32,200,76,67
1440 DATA 81,160,0,185,144,6,145,204,152
,201,16,240,4,200,76,83,81,189,64,6
1445 DATA 201,4,208,5,169,0,157,64,6,56,
189,72,6,233,1,157,72,6,76,199
1450 DATA 81,189,72,6,221,247,80,240,245
,189,64,6,24,105,1,157,64,6,201,3
1460 DATA 240,20,201,4,240,16,160,0,185,
128,6,145,204,152,201,16,240,32,200,76
1480 DATA 148,81,160,0,185,144,6,145,204
,152,201,16,240,4,200,76,164,81,189,64
1490 DATA 6,201,4,208,5,169,0,157,64,6,2
4,189,72,6,105,1,157,72,6
1495 DATA 24,165,204,105,3,133,204,76,7,
80
1500 DATA 0,0,0,0,14,14,31,30,14,2,2,0,0
,0,0,0
1510 DATA 0,0,0,0,14,14,31,15,14,8,8,0,0
,0,0,0
1520 DATA 173,72,6,170,169,0,157,254,153
,157,8,154,173,73,6,170,169,0,157,126,15
4,157,136,154,96
1530 DATA 104,24,173,74,1,201,150,176,18
,173,101,3,109,74,1,141,101,3,141,2
1540 DATA 208,141,3,208,76,99,117,56,173
,74,1,233,150,141,112,1,173,101,3,237
1550 DATA 112,1,141,101,3,141,2,208,141,
3,208,173,77,1,170,160,0,185,94,1
1560 DATA 157,0,155,232,200,192,9,240,3,
76,105,117,173,78,1,170,160,0,185,94
1570 DATA 1,157,128,155,232,200,192,10,2
40,3,76,126,117,173,75,1,24,201,150,176
1580 DATA 12,173,77,1,109,75,1,141,77,1,
76,179,117,56,173,75,1,233,150,141
1590 DATA 112,1,173,77,1,237,112,1,141,7
7,1,56,173,76,1,237,111,1,16,53
1591 DATA 56,173,111,1,237,76,1,205,108,
1,144,86,24,173,110,1,201,150,176,12
1592 DATA 173,78,1,109,110,1,141,78,1,76
,72,118,173,110,1,56,233,150,141,112
1593 DATA 1,173,78,1,237,112,1,141,78,1,
76,72,118,238,111,1,173,109,1,24
1594 DATA 201,150,176,12,173,78,1,109,10
9,1,141,78,1,76,72,118,56,173,109,1
1595 DATA 233,150,141,112,1,173,78,1,56,
237,112,1,141,78,1,76,72,118,238,111
1596 DATA 1,173,75,1,24,201,150,176,13,1
73,78,1,24,109,75,1,141,78,1,76
1597 DATA 72,118,56,173,75,1,233,150,141
,112,1,173,78,1,237,112,1,141,78,1
1598 DATA 76,185,86
1600 DATA 104,173,080,001,201,000,208,01
2,173,132,002,201,000,208,057,169,000,23
4,234,234
1610 DATA 160,000,173,072,006,170,024,17
3,080,001,201,003,176,041,173,101,003,20
5,080,006
1620 DATA 176,015,185,123,001,157,255,15
3,232,200,192,009,240,098,076,134,118,18
5,114,001
1630 DATA 157,255,153,232,200,192,010,24
0,083,076,149,118,076,093,120,024,173,08
0,001,201
```



```
1640 DATA 005,176,038,173,101,003,205,08
0,006,176,015,185,141,001,157,255,153,23
2,200,192
1650 DATA 010,240,049,076,183,118,185,13
2,001,157,255,153,232,200,192,009,240,03
4,076,198
1660 DATA 118,024,173,080,001,201,007,17
6,015,185,150,001,157,255,153,232,200,19
2,009,240
1670 DATA 011,076,221,118,169,000,141,08
0,001,076,093,120,238,080,001,173,082,00
1,201,001
1680 DATA 240,166,173,079,001,201,002,20
8,003,076,094,120,173,072,006,024,205,07
7,001,176
1690 DATA 224,173,077,001,056,237,072,00
6,024,201,007,176,212,056,173,080,006,23
7,101,003
1700 DATA 016,015,056,173,101,003,237,08
0,006,024,201,012,176,191,076,058,119,02
4,201,012
1710 DATA 176,183,056,173,077,001,237,07
8,001,024,201,012,176,027,169,001,141,08
2,001,169
1720 DATA 000,141,083,001,141,090,001,14
1,008,210,169,010,032,064,086,141,001,21
0,076,100
1730 DATA 119,076,093,120,173,120,002,20
1,006,208,008,169,003,141,074,001,076,18
9,119,201
1740 DATA 007,208,008,169,002,141,074,00
1,076,189,119,201,005,208,008,169,001,14
1,074,001
1750 DATA 076,189,119,201,010,208,008,16
9,153,141,074,001,076,189,119,201,011,20
8,008,169
1760 DATA 152,141,074,001,076,189,119,20
1,009,208,008,169,151,141,074,001,076,18
9,119,201
1770 DATA 015,208,005,169,001,141,090,00
1,169,000,141,074,001,056,173,077,001,23
7,078,001
1780 DATA 201,012,208,005,162,000,076,03
2,120,201,011,208,005,162,006,076,032,12
0,201,010
1790 DATA 208,005,162,012,076,032,120,20
1,009,208,005,162,018,076,032,120,201,00
8,208,005
1800 DATA 162,024,076,032,120,201,007,20
8,005,162,030,076,032,120,201,006,208,00
5,162,036
1810 DATA 076,032,120,201,005,208,005,16
2,042,076,032,120,201,004,208,005,162,04
8,076,032
1820 DATA 120,201,003,208,005,162,054,07
6,032,120,162,060,189,240,085,141,076,00
1,232,189
1830 DATA 240,085,141,109,001,232,189,24
0,085,141,110,001,232,189,240,085,141,10
8,001,232
1840 DATA 189,240,085,141,091,001,232,18
9,240,085,141,075,001,173,090,001,201,00
1,240,003
1850 DATA 076,093,120,162,066,169,000,14
1,090,001,076,032,120,096,173,077,001,02
4,205,072
1860 DATA 006,176,246,173,072,006,056,23
7,077,001,024,201,007,176,234,056,173,08
0,006,237
1870 DATA 101,003,016,015,056,173,101,00
3,237,080,006,024,201,012,176,213,076,14
4,120,024
1880 DATA 201,012,176,205,056,173,077,00
1,237,078,001,024,201,012,176,027,169,00
1,141,082
1890 DATA 001,169,000,141,083,001,141,09
0,001,141,008,210,169,010,032,064,086,14
1,001,210
1900 DATA 076,186,120,076,093,120,173,12
0,002,201,006,208,008,169,001,141,074,00
1,076,019
1910 DATA 121,201,007,208,008,169,002,14
1,074,001,076,019,121,201,005,208,008,16
9,003,141
1920 DATA 074,001,076,019,121,201,010,20
8,008,169,151,141,074,001,076,019,121,20
1,011,208
1930 DATA 008,169,152,141,074,001,076,01
9,121,201,009,208,008,169,153,141,074,00
1,076,019
1940 DATA 121,201,015,208,005,169,001,14
1,090,001,169,000,141,074,001,056,173,07
7,001,237
1950 DATA 078,001,201,012,208,005,162,00
0,076,118,121,201,011,208,005,162,006,07
6,118,121
1960 DATA 201,010,208,005,162,012,076,11
8,121,201,009,208,005,162,018,076,118,12
1,201,008
1970 DATA 208,005,162,024,076,118,121,20
1,007,208,005,162,030,076,118,121,201,00
6,208,005
1980 DATA 162,036,076,118,121,201,005,20
8,005,162,042,076,118,121,201,004,208,00
5,162,048
1990 DATA 076,118,121,201,003,208,005,16
2,054,076,118,121,162,060,189,084,086,14
1,076,001
2000 DATA 232,189,084,086,141,109,001,23
2,189,084,086,141,110,001,232,189,084,08
6,141,108
2010 DATA 001,232,189,084,086,141,091,00
1,232,189,084,086,141,075,001,173,090,00
1,201,001
2020 DATA 240,003,076,093,120,162,066,16
9,000,141,090,001,076,118,121
2030 DATA 112,119,39,120,112,112,88,136,
140,14,238,228,30,14,14,26,17,49
2040 DATA 112,112,32,112,112,112,88,136,
140,14,14,4,14,14,14,26,17,49
2045 DATA 28,28,8,62,62,28,20,20,54
2050 DATA 104,173,081,001,201,000,208,01
2,173,133,002,201,000,208,057,169,000,23
4,234,234
2060 DATA 160,000,173,073,006,170,024,17
3,081,001,201,003,176,041,173,101,003,20
5,081,006
2070 DATA 176,015,185,123,001,157,127,15
4,232,200,192,009,240,098,076,226,086,18
5,114,001
```



```
2080 DATA 157,127,154,232,200,192,009,24
0,083,076,241,086,076,185,088,024,173,08
1,001,201
2090 DATA 005,176,038,173,101,003,205,08
1,006,176,015,185,141,001,157,127,154,23
2,200,192
2100 DATA 009,240,049,076,019,087,185,13
2,001,157,127,154,232,200,192,009,240,03
4,076,034
2110 DATA 087,024,173,081,001,201,007,17
6,015,185,150,001,157,127,154,232,200,19
2,009,240
2120 DATA 011,076,057,087,169,000,141,08
1,001,076,185,088,238,081,001,173,083,00
1,201,001
2130 DATA 240,166,173,079,001,201,002,24
0,003,076,188,088,173,073,006,024,205,07
7,001,176
2140 DATA 224,173,077,001,056,237,073,00
6,024,201,007,176,212,056,173,081,006,23
7,101,003
2150 DATA 016,015,056,173,101,003,237,08
1,006,024,201,006,176,191,076,150,087,02
4,201,006
2160 DATA 176,183,056,173,077,001,237,07
8,001,024,201,012,176,027,169,001,141,08
3,001,169
2170 DATA 000,141,082,001,141,090,001,14
1,008,210,169,010,032,064,086,141,001,21
0,076,192
2180 DATA 087,076,185,088,173,121,002,20
1,006,208,008,169,003,141,074,001,076,02
5,088,201
2190 DATA 007,208,008,169,002,141,074,00
1,076,025,088,201,005,208,008,169,001,14
1,074,001
2200 DATA 076,025,088,201,010,208,008,16
9,153,141,074,001,076,025,088,201,011,20
8,008,169
2210 DATA 152,141,074,001,076,025,088,20
1,009,208,008,169,151,141,074,001,076,02
5,088,201
2220 DATA 015,208,005,169,001,141,090,00
1,169,000,141,074,001,056,173,077,001,23
7,078,001
2230 DATA 201,012,208,005,162,000,076,12
4,088,201,011,208,005,162,006,076,124,08
8,201,010
2240 DATA 208,005,162,012,076,124,088,20
1,009,208,005,162,018,076,124,088,201,00
8,208,005
2250 DATA 162,024,076,124,088,201,007,20
8,005,162,030,076,124,088,201,006,208,00
5,162,036
2260 DATA 076,124,088,201,005,208,005,16
2,042,076,124,088,201,004,208,005,162,04
8,076,124
2270 DATA 088,201,003,208,005,162,054,07
6,124,088,162,060,189,240,085,141,076,00
1,232,189
2280 DATA 240,085,141,109,001,232,189,24
0,085,141,110,001,232,189,240,085,141,10
8,001,232
2290 DATA 189,240,085,141,091,001,232,18
9,240,085,141,075,001,173,090,001,201,00
1,240,003
2300 DATA 076,185,088,162,066,169,000,14
1,090,001,076,124,088,076,017,090,173,07
7,001,024
2310 DATA 205,073,006,176,244,173,073,00
6,056,237,077,001,024,201,007,176,232,05
6,173,081
2320 DATA 006,237,101,003,016,015,056,17
3,101,003,237,081,006,024,201,006,176,21
1,076,238
2330 DATA 088,024,201,006,176,203,056,17
3,077,001,237,078,001,024,201,012,176,02
7,169,001
2340 DATA 141,083,001,169,000,141,082,00
1,141,090,001,141,008,210,169,010,032,06
4,086,141
2350 DATA 001,210,076,024,089,076,185,08
8,173,121,002,201,006,208,008,169,001,14
1,074,001
2360 DATA 076,113,089,201,007,208,008,16
9,002,141,074,001,076,113,089,201,005,20
8,008,169
2370 DATA 003,141,074,001,076,113,089,20
1,010,208,008,169,151,141,074,001,076,11
3,089,201
2380 DATA 011,208,008,169,152,141,074,00
1,076,113,089,201,009,208,008,169,153,14
1,074,001
2390 DATA 076,113,089,201,015,208,005,16
9,001,141,090,001,169,000,141,074,001,05
6,173,077
2400 DATA 001,237,078,001,201,012,208,00
5,162,000,076,212,089,201,011,208,005,16
2,006,076
2410 DATA 212,089,201,010,208,005,162,01
2,076,212,089,201,009,208,005,162,018,07
6,212,089
2420 DATA 201,008,208,005,162,024,076,21
2,089,201,007,208,005,162,030,076,212,08
9,201,006
2430 DATA 208,005,162,036,076,212,089,20
1,005,208,005,162,042,076,212,089,201,00
4,208,005
2440 DATA 162,048,076,212,089,201,003,20
8,005,162,054,076,212,089,162,060,189,08
4,086,141
2450 DATA 076,001,232,189,084,086,141,10
9,001,232,189,084,086,141,110,001,232,18
9,084,086
2460 DATA 141,108,001,232,189,084,086,14
1,091,001,232,189,084,086,141,075,001,17
3,090,001
2470 DATA 201,001,240,003,076,185,088,16
2,066,169,000,141,090,001,076,212,089,07
6,093,118
2480 DATA 141,0,210,169,0,141,111,1,169,
10,96
2500 DATA 224,0,208,12,173,80,1,201,0,24
0,17,104,104,76,241,80,173,81,1,201,0,24
0,5,104,104
2510 DATA 76,241,80,76,4,81
2520 DATA 104,174,072,006,160,000,173,11
3,001,201,000,208,015,185,104,091,157,25
5,153,232
```



```
2530 DATA 200,192,009,208,244,076,007,09
2,201,001,208,015,185,113,091,157,255,15
3,232,200
2540 DATA 192,009,208,244,076,007,092,18
5,122,091,157,255,153,232,200,192,009,20
8,244,238
2550 DATA 113,001,096
2560 DATA 56,56,144,124,58,59,43,72,216,
3,3,57,57,146,124,56,104,200,14,14,4,62,
22,102,102,5,5
2600 DATA 104,174,072,006,160,000,173,11
3,001,201,000,208,015,185,134,091,157,25
5,153,232
2610 DATA 200,192,009,208,244,076,107,09
2,201,001,208,015,185,143,091,157,255,15
3,232,200
2620 DATA 192,009,208,244,076,107,092,18
5,152,091,157,255,153,232,200,192,009,20
8,244,238
2630 DATA 113,001,096
2640 DATA 028,028,009,062,220,220,020,01
8,027,192,192,156,156,073,062,028,030,01
9,115,115
2650 DATA 036,120,112,112,120,072,108
2700 DATA 104,174,073,006,160,000,173,11
3,001,201,000,208,015,185,104,091,157,12
7,154,232
2710 DATA 200,192,009,208,244,076,207,09
2,201,001,208,015,185,113,091,157,127,15
4,232,200
2720 DATA 192,009,208,244,076,207,092,18
5,122,091,157,127,154,232,200,192,009,20
8,244,238
2730 DATA 113,001,096
2750 DATA 104,174,073,006,160,000,173,11
3,001,201,000,208,015,185,134,091,157,12
7,154,232
2760 DATA 200,192,009,208,244,076,051,09
3,201,001,208,015,185,143,091,157,127,15
4,232,200
2770 DATA 192,009,208,244,076,051,093,18
5,152,091,157,127,154,232,200,192,009,20
8,244,238
2780 DATA 113,001,096
2800 DATA 104,173,075,001,205,107,001,24
0,005,169,000,141,087,001,169,000,234,23
4,234,173
2801 DATA 082,001,141,107,001,173,252,00
2,201,033,208,031,169,255,141,252,002,16
0,000,162
2802 DATA 000,232,224,250,208,251,200,19
2,020,208,244,173,252,002,201,033,208,24
9,169,255
2803 DATA 141,252,002,024,173,078,001,20
5,077,001,176,003,076,172,094,173,087,00
1,201,001
2804 DATA 208,008,169,003,141,086,001,07
6,172,094,238,087,001,169,017,141,000,21
0,169,069
2805 DATA 141,001,210,173,007,208,201,00
0,208,008,169,001,141,086,001,076,172,09
4,173,093
2806 DATA 001,201,001,240,050,201,002,24
0,046,201,003,240,042,201,004,240,038,17
3,075,001
2807 DATA 024,201,140,176,019,169,067,02
4,205,077,001,176,003,076,149,094,169,00
2,141,086
2808 DATA 001,076,172,094,024,173,077,00
1,201,064,176,240,076,149,094,173,075,00
1,024,201
2809 DATA 140,176,019,024,173,077,001,20
1,081,176,003,076,149,094,169,001,141,08
6,001,076
2810 DATA 172,094,024,169,049,205,077,00
1,176,240,076,149,094,173,077,001,056,23
3,001,141
2811 DATA 078,001,169,000,141,111,001,14
1,093,001,173,091,001,141,076,001,169,00
0,141,030
2812 DATA 208,076,049,117
2850 DATA 104,169,000,141,001,210,024,17
3,077,001,201,100,176,024,169,030,205,07
7,001,176
2860 DATA 017,173,101,003,201,195,176,01
0,169,050,205,101,003,176,003,076,026,09
5,169,007
2870 DATA 141,086,001,076,191,095,173,07
5,001,201,002,240,012,201,003,240,022,20
1,152,240
2880 DATA 036,201,153,240,046,173,077,00
1,201,066,240,057,201,067,240,053,076,19
1,095,173
2890 DATA 077,001,201,066,240,043,201,06
7,240,039,201,068,240,035,076,191,095,17
3,077,001
2900 DATA 201,065,240,025,201,064,240,02
1,076,191,095,173,077,001,201,065,240,01
1,201,064
2910 DATA 240,007,201,063,240,003,076,19
1,095,056,173,077,001,237,078,001,141,11
2,001,024
2920 DATA 169,004,205,112,001,176,003,07
6,191,095,173,112,001,201,004,208,049,02
4,173,010
2930 DATA 210,201,200,176,008,169,004,14
1,086,001,076,191,095,169,000,141,000,21
0,169,010
2940 DATA 141,001,210,169,003,141,111,00
1,173,093,001,201,000,208,003,076,191,09
5,169,006
2950 DATA 141,086,001,076,191,095,169,00
5,141,086,001,076,193,93
2990 DATA 104,162,0,169,0,157,0,154,224,
255,240,4,232,76,27,96,162,0,169,0
3000 DATA 157,255,154,232,224,255,208,24
6,96
4500 DATA 0,2,4,0,6,3,0,2,4,0,5,3,0,2,4,
0,4,3,0,2,4,0,4,3,0,1,3,1,5,2,0,2,4,2,5,
3,3,2,4,3,5,3,4,1,3,6,5,2
4501 DATA 6,1,3,4,5,2,5,1,3,5,5,2,5,1,3,
6,3,2,12,1,4,2,6,2,0,154,152,0,6,153,0,1
54,152,0,5,153
4502 DATA 0,154,152,0,4,153,0,154,152,0,
4,153,0,153,151,1,5,152,0,154,152,2,5,15
3,3,154,152
4503 DATA 3,5,153,4,153,151,6,5,152,6,15
3,151,5,5,152,5,153,151,5,5,152,5,153,15
1,6,3,152,12,153,150,2,6,152
```



```
7010 DATA 125,40,125,35,112,5,100,40,125
,40,83,80,100,40,125,40
7020 DATA 93,40,83,18,93,18,100,18,93,18
,83,40
7025 DATA 112,18,125,18,112,18,100,18,11
2,40,168,40
7030 DATA 125,40,125,35,112,5,100,40,125
,40,83,80,100,40,125,40
7035 DATA 93,18,83,20,100,22,93,24,112,8
0,125,20,125,120
8500 DATA 104,024,169,035,205,077,001,17
6,003,076,009,100,234,173,010,210,201,12
7,176,056
8510 DATA 169,000,141,111,001,169,002,14
1,075,001,169,001,141,109,001,169,003,14
1,110,001
8520 DATA 169,006,141,076,001,169,004,14
1,108,001,024,173,101,003,201,123,176,00
9,173,167
8530 DATA 097,141,074,001,076,009,100,17
3,167,097,024,105,150,076,019,100,169,00
0,141,111
8540 DATA 001,169,003,141,075,001,169,00
2,141,109,001,169,004,141,110,001,169,00
5,141,086
8550 DATA 001,169,002,141,108,001,076,20
6,099,076,237,094,001,076,206,099,076,23
7,094,141
8560 DATA 074,001,169,000,141,082,001,07
6,009,100
9000 DATA 104,024,173,010,210,201,127,17
6,008,169,001,141,167,097,076,166,101,16
9,002,141
9010 DATA 167,097,173,166,097,201,148,20
8,249,162,000,160,000,200,192,001,208,25
1,232,236
9020 DATA 165,097,208,243,076,237,094,09
7,208,243,076,237,094
9030 DATA 169,148,141,166,97,169,64,141,
16,0,141,14,210,162,253,154,169,183,72,1
69,84,72,169,4,32,182,187
9040 DATA 169,255,76,4,187
```

TOP-TENNIS – TEIL ZWEI

```
1 REM Bernard Dalle
2 REM Weinbergstr.23
3 REM CH-8280 KREUZLINGEN
4 REM
10 POKE 25599,76:POKE 25602,3:POKE 25541
,9:POKE 20623,5:POKE 20714,5
20 POKE 24462,150:POKE 24218,0:POKE 2402
0,75:POKE 24998,148
25 POSITION 11,6:? "NOW LOADING PART 3"
26 POKE 30511,8:POKE 30519,8:POKE 22411,
8:POKE 22419,8
30 POKE 764,33:A=USR(26053)
```

TOP-TENNIS – TEIL DREI

```
5 CLR :? "K":GOSUB 7000
8 POKE 54279,152:POKE 24462,150:POKE 242
18,0:POKE 24020,75:POKE 24998,148:GRAPHI
CS 15:POKE 559,0
13 POKE 33008,112:POKE 32854,2:POKE 3300
9,112:POKE 752,1:POKE 342,0
15 POKE 87,0:POSITION 0,1:? #6;"
     GAMES:          SETS:":POKE 87,15
30 COLOR 2:PLOT 30,149:DRAWTO 129,149:DR
AWTO 119,50:DRAWTO 40,50:DRAWTO 30,149
35 POKE 765,1:POSITION 40,51:XIO 18,#6,0
,0,"S:":PLOT 80,148:PLOT 80,51
40 PLOT 32,128:DRAWTO 127,128:PLOT 38,70
:DRAWTO 120,70:PLOT 80,70:DRAWTO 80,128
60 PLOT 10,30:DRAWTO 10,15:DRAWTO 149,15
:DRAWTO 149,30:DRAWTO 10,30:POSITION 10,
16:COLOR 2:XIO 18,#6,0,0,"S:"
61 PLOT 5,40:DRAWTO 10,15:DRAWTO 10,30:D
RAWTO 5,55:DRAWTO 5,40:PLOT 149,15:DRAWT
O 154,40
63 DRAWTO 154,55:DRAWTO 149,30:FOR X=10
TO 145 STEP 6
64 PLOT X,15:DRAWTO X,10:DRAWTO X+4,10:D
RAWTO X+4,15:PLOT X+2,9:PLOT X+1,8:DRAWT
O X+3,8:DRAWTO X+3,6
65 DRAWTO X+1,6:DRAWTO X+1,8:NEXT X
70 FOR X=27 TO 132 STEP 3:PLOT X,90:DRAW
TO X,100:NEXT X:FOR X=90 TO 100 STEP 2:P
LOT 27,X:DRAWTO 132,X:NEXT X
75 FOR X=0 TO 3:PLOT 24+X,90:DRAWTO 24+X
,100:PLOT 132+X,90:DRAWTO 132+X,100:NEXT
 X
76 COLOR 0:FOR X=33 TO 126 STEP 3:PLOT X
,101:DRAWTO X,105:NEXT X:FOR X=101 TO 10
5 STEP 2:PLOT 33,X
77 DRAWTO 126,X:NEXT X
90 POKE 709,14:POKE 708,212:POKE 710,224
:POKE 712,224:POKE 704,220:POKE 705,158:
POKE 706,1:POKE 707,238
155 POKE 53258,1:POKE 53259,1:POKE 559,4
6:POKE 53277,3:POKE 337,0:POKE 336,0:POK
E 623,1:GOTO 400
300 POKE 343,0:POKE 77,0:IF HY THEN 4020
301 A=USR(20730,0,39420+PEEK(1608)):B=US
R(20730,1,39548+PEEK(1609)):A=USR(26000)
:IF PEEK(342) THEN 310
303 GOTO 301
310 POKE 53250,0:POKE 53761,0:ON PEEK(34
2) GOTO 1610,312,1620,1610,1610,1630,320
312 IF PEEK(333)<99 AND PEEK(333)>72 OR
PEEK(333)<55 AND PEEK(333)>40 THEN POKE
343,0:POKE 342,0:POKE 367,0:GOTO 300
313 GOTO 1610
320 IF PEEK(343)=1 THEN 1620
321 GOTO 1610
322 POKE 20725,30:POKE 20726,70:POKE 207
27,60:POKE 20728,100:RETURN
323 POKE 20725,70:POKE 20726,30:POKE 207
27,100:POKE 20728,60:RETURN
329 POKE 39811+PEEK(334),0
330 FOR X=0 TO 8:POKE 39813+PEEK(334)+X,
32:POKE 39812+PEEK(334)+X,0:FOR G=1 TO 1
0:NEXT G:NEXT X:IF HY THEN 4062
331 POKE 349,1:POKE 342,5:GOTO 310
366 IF X=15 THEN 330
400 POKE 1608,88:POKE 334,91:POKE 1616,1
26:POKE 53251,125:POKE 333,91:RESTORE :K
L=0:PO=128:M=0:GOTO 1540
401 POKE 1608,88:POKE 334,91:POKE 1616,1
20:POKE 53251,119:POKE 333,91:RESTORE :K
```



```
L=100:PO=122:M=0:GOTO 1545
402 POKE 1608,37:POKE 334,39:POKE 1616,1
24:POKE 53251,120:POKE 333,39:RESTORE 46
0:KL=200:PO=119:M=100:GOTO 1550
403 POKE 1608,37:POKE 334,39:POKE 1616,1
30:POKE 53251,126:POKE 333,39:RESTORE 46
0:KL=300:PO=125:M=100:GOTO 1555
404 GOSUB 321+PEEK(335):GOSUB 470+J:POKE
 53248+J,PO-2+M*0.04
405 SOUND 0,0,0,0:FOR X=0 TO 3:POKE 3981
2+PEEK(334),0:POKE 39811+PEEK(334),32
406 GOSUB 470+J
410 POKE 334,PEEK(334)-1:NEXT X
412 IF PEEK(53279)=6 THEN GRAPHICS 0:POK
E 710,144:CLR :GOSUB 6000:GOTO 8
415 FOR X=0 TO 8:READ C:POKE 39423+J*128
+PEEK(1608+J)+X,C:IF STRIG(J)=0 THEN POP
 :GOTO 650
416 GOSUB 470+J
420 NEXT X:FOR X=0 TO 8:READ C:POKE 3942
3+J*128+PEEK(1608+J)+X,C:IF STRIG(J)=0 T
HEN POP :GOTO 650
424 GOSUB 470+J
425 NEXT X:FOR X=0 TO 3:POKE 39808+PEEK(
334)+3,0:POKE 39808+PEEK(334)+4,32
430 POKE 334,PEEK(334)+1:GOSUB 470+J:NEX
T X
435 SOUND 0,10,8,6:RESTORE 450+M/10
440 GOTO 404
450 DATA 7,7,2,30,6,6,5,9,27,7,7,2,6,30,
6,5,9,27
460 DATA 224,224,64,248,224,224,160,144,
216,224,224,64,224,248,224,160,144,216
470 A=USR(20730,1,39548+PEEK(1609)):RETU
RN
471 A=USR(20730,0,39420+PEEK(1608)):RETU
RN
500 POKE 1609,88:POKE 334,91:POKE 1617,1
26:POKE 53251,125:POKE 333,91:RESTORE :K
L=0:PO=128:M=0:GOTO 1560
501 POKE 1609,88:POKE 334,91:POKE 1617,1
20:POKE 53251,119:POKE 333,91:RESTORE :K
L=100:PO=122:M=0:GOTO 1565
502 POKE 1609,37:POKE 334,39:POKE 1617,1
24:POKE 53251,120:POKE 333,39:RESTORE 46
0:KL=200:PO=119:M=100:GOTO 1570
503 POKE 1609,37:POKE 334,39:POKE 1617,1
30:POKE 53251,126:POKE 333,39:RESTORE 46
0:KL=300:PO=125:M=100:GOTO 1575
505 GOTO 404
650 POKE 369,0:POKE 53248+J,PO:A=USR(235
00+M+J*200)
652 DEL=0:FOR X=0 TO 16:GOSUB 470+J:POKE
 39812+PEEK(334),0:POKE 39811+PEEK(334),
32
655 FOR G=1 TO INT(DEL/2)+1:NEXT G:DEL=D
EL+1:POKE 334,PEEK(334)-1:NEXT X
660 FOR X=0 TO 20:POKE 39811+PEEK(334),0
:POKE 39812+PEEK(334),32
665 FOR G=1 TO 5:NEXT G:POKE 334,PEEK(33
4)+1:IF STRIG(J)=0 THEN POP :POKE 39812+
PEEK(334),0:GOTO 700+KL
667 IF X>11 THEN POP :GOTO 329
670 GOSUB 470+J:NEXT X
700 GOSUB 700+STICK(J)
701 A=USR(23500+J*200)
702 GOSUB 722:POKE 349,1
703 A=USR(23500+J*200):POKE 1616+J,128:P
OKE 869,123:SOUND 0,10,8,15
704 GOTO 804
705 POKE 330,0:POKE 331,152:POKE 365,153
:POKE 366,151:POKE 367,0:RETURN
706 POKE 330,0:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
707 POKE 330,0:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
709 POKE 330,152:POKE 331,152:POKE 365,1
53:POKE 366,151:POKE 367,0:RETURN
710 POKE 330,152:POKE 331,153:POKE 365,1
54:POKE 366,151:POKE 367,0:RETURN
711 POKE 330,152:POKE 331,153:POKE 365,1
54:POKE 366,151:POKE 367,0:RETURN
713 POKE 330,151:POKE 331,152:POKE 365,1
53:POKE 366,151:POKE 367,0:RETURN
714 POKE 330,151:POKE 331,153:POKE 365,1
54:POKE 366,151:POKE 367,0:RETURN
715 POKE 330,151:POKE 331,152:POKE 365,1
53:POKE 366,151:POKE 367,0:RETURN
722 IF PEEK(331)>140 THEN 726
723 POKE 354,0:POKE 355,32:POKE 348,1
724 POKE 364,0:POKE 332,2:IF X<4 THEN PO
KE 364,4:POKE 332,2:IF PEEK(331)=2 THEN
POKE 364,1
725 GOTO 728
726 POKE 354,32:POKE 355,0:POKE 349,0:PO
KE 364,0:POKE 332,2
727 IF X<6 THEN POKE 364,4:POKE 332,2
728 POKE 347,7:RETURN
800 GOSUB 800+STICK(J)
801 A=USR(23500+J*200)
802 GOSUB 722:POKE 349,2
803 A=USR(23500+J*200):POKE 1616+J,PO:PO
KE 869,123:SOUND 0,10,8,15:POKE PM+896+P
EEK(334)+3,0
804 POKE 347,5:POKE 338,ABS(J-1):POKE 33
9,J:GOTO 300
805 POKE 330,2:POKE 331,152:POKE 365,153
:POKE 366,151:POKE 367,0:RETURN
806 POKE 330,2:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
807 POKE 330,2:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
809 POKE 330,0:POKE 331,152:POKE 365,153
:POKE 366,151:POKE 367,0:RETURN
810 POKE 330,0:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
811 POKE 330,0:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
813 POKE 330,1:POKE 331,152:POKE 365,153
:POKE 366,151:POKE 367,0:RETURN
814 POKE 330,1:POKE 331,153:POKE 365,154
:POKE 366,151:POKE 367,0:RETURN
815 POKE 330,1:POKE 331,152:POKE 365,153
:POKE 366,151:POKE 367,0:RETURN
900 GOSUB 900+STICK(J)
901 A=USR(23600+J*200)
902 GOSUB 722:POKE 349,3
903 A=USR(23600+J*200):POKE 1616+J,PO:PO
KE 869,123:SOUND 0,10,8,15:POKE PM+768+P
EEK(333)+2,0
```

```
904 GOTO 804
905 POKE 330,2:POKE 331,3:POKE 365,2:POK
E 366,5:POKE 367,0:RETURN
906 POKE 330,2:POKE 331,2:POKE 365,1:POK
E 366,3:POKE 367,0:RETURN
907 POKE 330,2:POKE 331,3:POKE 365,2:POK
E 366,5:POKE 367,0:RETURN
909 POKE 330,0:POKE 331,3:POKE 365,2:POK
E 366,5:POKE 367,0:RETURN
910 POKE 330,0:POKE 331,2:POKE 365,1:POK
E 366,3:POKE 367,0:RETURN
911 POKE 330,0:POKE 331,3:POKE 365,2:POK
E 366,5:POKE 367,0:RETURN
913 POKE 330,1:POKE 331,3:POKE 365,2:POK
E 366,5:POKE 367,0:RETURN
914 POKE 330,1:POKE 331,2:POKE 365,1:POK
E 366,3:POKE 367,0:RETURN
915 POKE 330,1:POKE 331,2:POKE 365,1:POK
E 366,3:POKE 367,0:RETURN
1000 GOSUB 1000+STICK(J)
1001 A=USR(23600+J*200)
1002 GOSUB 722:POKE 349,4
1003 A=USR(23600+J*200):POKE 1616+J,PO:P
OKE 869,123:SOUND 0,10,8,15:POKE PM+768+
PEEK(333)+2,0
1004 GOTO 804
1005 POKE 330,0:POKE 331,3:POKE 365,2:PO
KE 366,5:POKE 367,0:RETURN
1006 POKE 330,0:POKE 331,2:POKE 365,1:PO
KE 366,3:POKE 367,0:RETURN
1007 POKE 330,0:POKE 331,3:POKE 365,2:PO
KE 366,5:POKE 367,0:RETURN
1009 POKE 330,152:POKE 331,3:POKE 365,2:
POKE 366,5:POKE 367,0:RETURN
1010 POKE 330,152:POKE 331,2:POKE 365,1:
POKE 366,3:POKE 367,0:RETURN
1011 POKE 330,152:POKE 331,3:POKE 365,2:
POKE 366,5:POKE 367,0:RETURN
1013 POKE 330,151:POKE 331,3:POKE 365,2:
POKE 366,5:POKE 367,0:RETURN
1014 POKE 330,151:POKE 331,2:POKE 365,1:
POKE 366,3:POKE 367,0:RETURN
1015 POKE 330,151:POKE 331,2:POKE 365,1:
POKE 366,3:POKE 367,0:RETURN
1500 IF (GAMEA+GAMEB)/2<>INT((GAMEA+GAME
B)/2) THEN CHANGE=3:RETURN
1510 RETURN
1520 IF CH=3 THEN CH=0:RETURN
1522 IF SER=0 THEN SER=100:RETURN
1530 SER=0:RETURN
1540 POKE 1617,96*ABS(HY-1):POKE 1609,35
:POKE 335,2:J=0:GOTO 404
1545 POKE 1617,150:POKE 1609,35:POKE 335
,2:J=0:GOTO 404
1550 POKE 1617,150:POKE 1609,90:POKE 335
,1:J=0:GOTO 404
1555 POKE 1617,96:POKE 1609,90:POKE 335,
1:J=0:GOTO 404
1560 POKE 1616,96:POKE 1608,35:POKE 335,
1:J=1:GOTO 404
1565 POKE 1616,150:POKE 1608,35:POKE 335
,1:J=1:GOTO 404
1570 POKE 1616,150:POKE 1608,90:POKE 335
,2:J=1:GOTO 404
1575 POKE 1616,96:POKE 1608,90:POKE 335,
2:J=1:GOTO 404
1610 IF PEEK(338)=0 THEN BA=1:BB=0:GOTO
1700
1615 BA=0:BB=1:GOTO 1700
1620 IF PEEK(338)=0 THEN BA=0:BB=1:GOTO
1700
1625 BB=0:BA=1:GOTO 1700
1630 POKE 342,0:BA=0:BB=0:GOTO 1750
1700 FOR X=15 TO 0 STEP -0.5:SOUND 0,20,
10,X:NEXT X
1704 IF PEEK(342)=1 THEN ? "
   OUT"
1715 IF PEEK(349)=0 THEN 1850
1720 IF NSER=0 THEN ? :? "              SE
COND SERVICE":NSER=1:BA=0:BB=0:GOTO 1850
1725 NSER=0:GOTO 1850
1750 ? "                    LET":IF NSER=0
THEN ? :? "                FIRST SERVICE":
GOTO 1800
1760 ? :? "              SECOND SERVICE"
1800 GOSUB 470:GOSUB 471:A=USR(24300)
1810 IF PEEK(342)=7 OR PEEK(342)=5 THEN
GOTO 1850
1820 GOTO 1800
1850 IF BA=0 AND BB=0 THEN 2000
1855 NSER=0:ADV=0:DEUCE=0:IF TIE=1 THEN
3400
1860 PINTA=PINTA+(BA=1):PINTB=PINTB+(BB=
1):CHANGE=2
1865 IF PINTA>3 THEN 1880
1870 IF PINTB>3 THEN 1880
1875 GOTO 2000
1880 IF PINTA-PINTB>1 THEN GAMEA=GAMEA+1
:PINTA=0:PINTB=0:CHANGE=1:GOSUB 1500:GOT
O 1920
1895 IF PINTB-PINTA>1 THEN GAMEB=GAMEB+1
:PINTA=0:PINTB=0:CHANGE=1:GOSUB 1500:GOT
O 1920
1900 IF PINTB-PINTA=0 THEN DEUCE=1:GOTO
2000
1910 IF PINTA>PINTB THEN ADV=1:GOTO 2000
1915 ADV=2:GOTO 2000
1920 IF GAMEA>5 THEN 3000
1930 IF GAMEB>5 THEN 3000
2000 POKE 87,0:IF TIE=1 THEN POSITION 1,
1:? #6;PINTA;"-";PINTB;"      ":GOTO 2040
2010 POA=15*(PINTA=1)+30*(PINTA=2)+40*(P
INTA=3)
2020 POB=15*(PINTB=1)+30*(PINTB=2)+40*(P
INTB=3)
2030 IF DEUCE=1 THEN POSITION 1,1:? #6;"
DEUCE";"     ":GOTO 2040
2032 IF ADV=1 THEN POSITION 1,1:? #6;"AD
V. PL.1":GOTO 2040
2034 IF ADV=2 THEN POSITION 1,1:? #6;"AD
V PL 2":GOTO 2040
2035 IF SER=100 THEN POSITION 1,1:? #6;P
OB;"-";POA;"      ":GOTO 2040
2036 POSITION 1,1:? #6;POA;"-";POB;"
  "
2040 POSITION 20,1:? #6;GAMEA;"-";GAMEB
2060 POSITION 33,1:? #6;SETA;"-";SETB
2070 IF CHANGE=1 OR CHANGE=3 THEN GOSUB
```



```
1520:POSX=0
2073 IF CHANGE=1 THEN IF POSY=2 THEN POS
Y=0:POSX=0:GOTO 2100
2074 IF CHANGE=1 THEN IF POSY=0 THEN POS
Y=2:POSX=0:GOTO 2100
2075 IF CHANGE=2 THEN IF POSX=1 THEN POS
X=0:CHANGE=0
2077 IF CHANGE=2 THEN CHANGE=0:IF POSX=0
 THEN POSX=1:GOTO 2100
2100 FOR G=1 TO 100:NEXT G
2105 ? CHR$(125):CHANGE=0:POKE 336,0:POK
E 337,0:POKE 342,0:POKE 343,0:A=USR(2460
0):GOTO 400+SER+POSY+POSX
3000 IF GAMEA-GAMEB>1 THEN SETA=SETA+1:G
AMEA=0:GAMEB=0:GOTO 3200
3010 IF GAMEA=GAMEB THEN TIE=1:? :? "
          TIE BREAK":POSX=1:GOTO 2000
3050 IF GAMEB-GAMEA>1 THEN SETB=SETB+1:G
AMEA=0:GAMEB=0:GOTO 3200
3070 GOTO 2000
3200 IF SETA=2 THEN V=1:GOTO 3300
3210 IF SETB=2 THEN V=2:GOTO 3300
3220 GOTO 2000
3300 FOR X=0 TO 3:POKE 53248+X,0:NEXT X
3310 GRAPHICS 0:POKE 710,0:POKE 752,1:PO
SITION 10,10:? "PLAYER ";V;" IS THE WINN
ER"
3320 FOR G=1 TO 500:NEXT G:GOTO 5
3400 PINTA=PINTA+(BA=1):PINTB=PINTB+(BB=
1):CHANGE=2:GOSUB 4000
3405 IF PINTA+PINTB=6 THEN CH=3:CHANGE=1
3410 IF PINTA>6 THEN 3440
3420 IF PINTB>6 THEN 3440
3430 GOTO 2000
3440 IF PINTA-PINTB>1 THEN GAMEA=0:GAMEB
=0:PINTA=0:PINTB=0:SETA=SETA+1:GOSUB 150
0:TIE=0:GOTO 3200
3450 IF PINTB-PINTA>1 THEN GAMEB=0:GAMEA
=0:PINTA=0:PINTB=0:SETB=SETB+1:TIE=0:GOS
UB 1500:GOTO 3200
3460 GOTO 2000
4000 IF (PINTA+PINTB)/2<>INT((PINTA+PINT
B)/2) THEN CHANGE=1:RETURN
4010 RETURN
4020 POKE 343,0:A=USR(25500):A=USR(26000
):POKE 338,0:IF PEEK(342) AND PEEK(342)<
>2 THEN 4060
4040 A=USR(20730,0,39420+PEEK(1608)):GOT
O 4020
4060 IF PEEK(342)>40 AND PEEK(342)<60 TH
EN POKE 367,0:GOTO 4065
4061 IF PEEK(342)<>5 AND PEEK(342)<>4 AN
D PEEK(342)<>7 THEN POKE 342,0:POKE 367,
0:GOTO 4065
4062 POKE 53250,0:FOR X=15 TO 0 STEP -1:
SOUND 0,20,2,X:NEXT X:POKE 342,0:A=USR(2
4600):GOTO 400
4065 POKE 342,0:POKE 334,PEEK(333)-1:A=U
SR(24600):GOTO 4020
6000 POKE 559,0:POSITION 4,8:? "OPTION"
:POSITION 4,13:? "SELECT":POSITION 10,1
8:? "press START to begin"
6001 POKE 752,1:A=USR(24600):FOR X=0 TO
3:POKE 53248+X,0:NEXT X:POKE 53278,0:POK
E 869,0:POKE 559,46
6002 POKE 710,144:FOR G=1 TO 100:NEXT G
6010 S=S+(PEEK(53279)=5):O=O+(PEEK(53279
)=3):IF S=3 THEN S=0
6015 IF O=2 THEN O=0
6016 GOSUB 6100+S*10:GOSUB 6200+O*10
6020 IF PEEK(53279)=6 THEN RETURN
6030 GOTO 6010
6100 POSITION 11,13:? "NOVICE  ":POKE 26
035,100:POKE 24997,50:RETURN
6110 POSITION 11,13:? "ADVANCED":POKE 26
035,100:POKE 24997,13:RETURN
6120 POSITION 11,13:? "EXPERT  ":POKE 26
035,1:POKE 24997,1:RETURN
6200 POSITION 11,8:? "2 PLAYERS GAME   "
:HY=0:RETURN
6210 POSITION 11,8:? "1 PLAYER PRACTICE"
:HY=1:RETURN
7000 A=PEEK(25000+X):B=PEEK(25001+X)
7007 SOUND 1,A,10,4:SOUND 0,A-1,10,4:FOR
 Q=1 TO B/1.6:NEXT Q:FOR G=1 TO 2:POKE 5
3761,0:POKE 53763,0:NEXT G
7008 X=X+2:IF X=70 THEN 6000
7010 GOTO 7000
```

Ende

Achtung! ... – Achtung! ... – Atari-Freunde aufgepaßt!!

Das Programm „Groove" ist jetzt auch auf DOS III lauffähig!!!

Leider ist uns beim Abdruck des Atari-Programms „Groove" ein Fehler unterlaufen. Denn in der abgedruckten Version ist dieses Programm nur auf DOS II lauffähig. Damit nun auch DOS III-Benutzer in den Genuß dieses tollen Programms gelangen können, hat unserer Programmierer Ottfried Schmidt die nachstehenden Änderungen vorgenommen:

In Teil 2 muß die Zeile **1890** folgendermaßen geändert werden:

1890 DATA 2,32,170,32,31,169,32,197,170,32,230,170,32, 189,168,0,0,0,0,0,69,0,0,0,2348

Die Zeile **160** muß lauten: **160 READ a**

Ansonsten wird beim Erstellen des Files GROOVE. EXE, wie in der Anleitung beschrieben, vorgegangen.

Um das Spiel zu starten, muß beim Einschalten des Computers die Taste „OPTION" gedrückt werden. Ist das DOS geladen, drücken Sie bitte die Taste „L". Als Dateinamen geben Sie bitte GROOVE:EXE ein. Ist das Programm geladen, so müssen Sie die Taste „G" betätigen. Auf die Frage nach der Startadresse geben Sie bitte die Zahl 4500 ein.

Das Programm wird nun gestartet und kann mit Joystick 1 gespielt werden.



Panzerschlacht

für den Atari 800 XL

PANZERSCHLACHT – ein wahrhaft explosives Spiel!

Viel Geschick und Reaktionsvermögen solltet Ihr schon besitzen, wenn der Panzer des Gegners ausgetrickst werden soll. Gelenkt wird mit dem Joystick (Port 1 = schwarzer Panzer, Port 2 = weißer Panzer), geschossen wird mit dem Feuerknopf. Allerdings könnt Ihr, wie sonst bei Panzern üblich, nicht auf Dauerfeuer umstellen.

Euer Fahrzeug kann in acht verschiedene Richtungen bewegt werden. Gewonnen hat derjenige Spieler, der den Panzer des anderen fünfzehnmal zerstören konnte. Die viereckigen Blöcke auf dem Spielfeld können weder überfahren noch abgeschossen werden.
Das Spielfeld ist nicht besonders groß, dafür sind die Panzer aber sehr schnell. Seid deshalb vorsichtig, damit der Gegenspieler nicht plötzlich hinter Euch ist . . . !
Hinweis zum Eintippen: Es werden keine Rem-Zeilen angesprungen, diese können somit vernachlässigt werden. (S.G.)

Hier werden Geschicklichkeit und schnelles Reaktionsvermögen gefordert

```
1 REM ******************
2 REM * PANZERSCHLACHT *
3 REM *   (C)1985 BY   *
4 REM *   ASKI--WARE   *
5 REM ******************
20 GOSUB 32000
50 LT=0:RT=0
110 GOSUB 30000
120 X0=2:Y0=3:X1=17:Y1=9:T0=0:T1=0
125 GOSUB 20000
130 GOTO 5000
999 REM ------BEWEGUNG PANZER 1------
1000 H0=S0+100:SOUND 0,250,6,12:COLOR 32
:PLOT X0,Y0:X0=X0+XL:Y0=Y0+YL:COLOR H0:P
LOT X0,Y0:RETURN
1099 REM -----BEWEGUNG PANZER 2------
1100 H1=S1+196:SOUND 1,240,6,12:COLOR 32
:PLOT X1,Y1:X1=X1+XR:Y1=Y1+YR:COLOR H1:P
LOT X1,Y1:RETURN
1199 REM -------PANZER 1 KAPUTT------
1200 IF S1<>15 THEN COLOR S1+196:PLOT X1
,Y1
1205 FOR I=255 TO 0 STEP -5:POKE 709,I:S
OUND 0,I,12,10:SOUND 3,I,6,10:NEXT I:FOR
 I=0 TO 3:SOUND I,0,0,0:NEXT I
1210 POP :RT=RT+1:GOTO 120
1299 REM -------PANZER 2 KAPUTT------
1300 IF S0<>15 THEN COLOR S0+100:PLOT X0
,Y0
1305 FOR I=255 TO 0 STEP -5:POKE 710,I:S
OUND 0,I,12,10:SOUND 2,I,6,10:NEXT I:FOR
 I=0 TO 3:SOUND I,0,0,0:NEXT I
1310 POP :LT=LT+1:GOTO 120
1999 REM -----SCHUSS ABGEGEBEN-------
2000 T0=1:P0=XL:Q0=YL:PX0=X0:QX0=Y0
2010 LOCATE PX0+P0,QX0+Q0,FL0
2020 IF FL0=32 OR FL0=164 THEN SOUND 2,P
X0+QX0,8,10:PX0=PX0+P0:QX0=QX0+Q0:COLOR
4:PLOT PX0,QX0:RETURN
2030 IF FL0=35 THEN T0=0:SOUND 2,0,0,0:R
ETURN
2040 IF FL0=248 THEN SOUND 2,200,8,12:CO
LOR 32:PLOT PX0+P0,QX0+Q0:T0=0:SOUND 2,0
,0,0:RETURN
2050 IF FL0=H1 THEN 1300
2999 REM ----------SCHUSSFLUG--------
3000 LOCATE PX0+P0,QX0+Q0,FL0
3010 IF FL0=32 OR FL0=164 THEN COLOR 32:
PLOT PX0,QX0:PX0=PX0+P0:QX0=QX0+Q0:COLOR
 4:PLOT PX0,QX0
3015 IF FL0=32 OR FL0=164 THEN SOUND 2,P
X0+QX0,8,10:RETURN
3020 IF FL0=35 THEN T0=0:COLOR 32:PLOT P
X0,QX0:SOUND 2,0,0,0:RETURN
3030 IF FL0=248 THEN SOUND 2,200,8,12:CO
LOR 32:PLOT PX0,QX0:PLOT PX0+P0,QX0+Q0:T
0=0:SOUND 2,0,0,0:RETURN
3040 IF FL0=H1 THEN COLOR 32:PLOT PX0,QX
0:T0=0:GOTO 1300
3499 REM -----SCHUSS ABGEGEBEN-------
3500 T1=1:P1=XR:Q1=YR:PX1=X1:QX1=Y1
3510 LOCATE PX1+P1,QX1+Q1,FL1
3520 IF FL1=32 OR FL1=4 THEN SOUND 3,PX1
+QX1,8,10:PX1=PX1+P1:QX1=QX1+Q1:COLOR 16
4:PLOT PX1,QX1:RETURN
3530 IF FL1=35 THEN T1=0:SOUND 3,0,0,0:R
ETURN
3540 IF FL1=248 THEN SOUND 3,200,8,10:CO
LOR 32:PLOT PX1+P1,QX1+Q1:T1=0:SOUND 3,0
,0,0:RETURN
3550 IF FL1=H0 THEN 1200
3999 REM ----------SCHUSSFLUG--------
4000 LOCATE PX1+P1,QX1+Q1,FL1
4010 IF FL1=32 OR FL1=4 THEN COLOR 32:PL
OT PX1,QX1:PX1=PX1+P1:QX1=QX1+Q1:COLOR 1
64:PLOT PX1,QX1
4015 IF FL1=32 OR FL1=4 THEN SOUND 3,PX1
+QX1,8,10:RETURN
4020 IF FL1=35 THEN T1=0:COLOR 32:PLOT P
X1,QX1:SOUND 3,0,0,0:RETURN
4030 IF FL1=248 THEN SOUND 3,200,8,12:CO
```

```
LOR 32:PLOT PX1,QX1:PLOT PX1+P1,QX1+Q1:T
1=0:SOUND 3,0,0,0:RETURN
4040 IF FL1=H0 THEN COLOR 32:PLOT PX1,QX
1:T1=0:GOTO 1200
4499 REM ----------SPIELENDE----------
4500 POSITION 2,4:? #6;"#### ##### ###  ####
"
4510 POSITION 2,5:? #6;"#   #  # # # #"
4520 POSITION 2,6:? #6;"###  #  # # # ###"
4530 POSITION 2,7:? #6;"#   #  # # # #"
4540 POSITION 2,8:? #6;"#### #  # ###  ####
"
4550 FOR I=0 TO 800:NEXT I:GOTO 50
4999 REM --------HAUPTSCHLEIFE--------
5000 PA=PA+1:IF PA>1 THEN PA=0
5002 IF PA=0 THEN 5100
5005 S0=STICK(0):IF S0=15 THEN SOUND 0,0
,0,0:GOTO 5090
5010 XL=0:YL=0
5020 XL=XL+(S0=5)+(S0=6)+(S0=7)-(S0=9)-(
S0=10)-(S0=11)
5030 YL=YL+(S0=5)+(S0=13)+(S0=9)-(S0=10)
-(S0=14)-(S0=6)
5040 LOCATE X0+XL,Y0+YL,F0:IF F0=32 THEN
 GOSUB 1000
5050 IF F0=248 THEN 1200
5090 IF STRIG(0)=0 AND T0=0 THEN GOSUB 2
000
5100 IF T0=1 THEN GOSUB 3000
5500 IF PA=1 THEN 5600
5505 S1=STICK(1):IF S1=15 THEN SOUND 1,0
,0,0:GOTO 5590
5510 XR=0:YR=0
5520 XR=XR+(S1=5)+(S1=6)+(S1=7)-(S1=9)-(
S1=10)-(S1=11)
5530 YR=YR+(S1=5)+(S1=13)+(S1=9)-(S1=10)
-(S1=14)-(S1=6)
5540 LOCATE X1+XR,Y1+YR,F1:IF F1=32 THEN
 GOSUB 1100
5550 IF F1=248 THEN 1300
5590 IF STRIG(1)=0 AND T1=0 THEN GOSUB 3
500
5600 IF T1=1 THEN GOSUB 4000
6000 GOTO 5000
19999 REM ----------SPIELFELD---------
20000 GRAPHICS 18:POKE 77,0:POKE 756,P:P
OKE 708,52:POKE 709,0:POKE 710,12:POKE 7
12,6:POKE 711,196
20010 POKE 711,INT((RND(0)*15)+1)*16+4:P
OKE 708,INT((RND(0)*15)+1)*16+2
20050 POSITION 8,0:? #6;LT:POSITION 10,0
:? #6;":":POSITION 11,0:? #6;RT
```

P A N Z E R S C H L A C H T — T H E E N D

```
20100 COLOR 35:PLOT 0,1:DRAWTO 19,1:DRAW
TO 19,11:DRAWTO 0,11:DRAWTO 0,1
20105 IF RT=15 OR LT=15 THEN POP :GOTO 4
500
20110 FOR I=0 TO 24
20120 F=INT(RND(0)*18)+1:G=INT(RND(0)*9)
+2:LOCATE F,G,FA:IF FA<>32 THEN 20120
20130 COLOR 35:PLOT F,G:NEXT I
20140 FOR I=0 TO 14
20150 F=INT(RND(0)*18)+1:G=INT(RND(0)*9)
+2:LOCATE F,G,FA:IF FA<>32 THEN 20150
20160 COLOR 248:PLOT F,G:NEXT I
20170 COLOR 113:PLOT X0,Y0:COLOR 210:PLO
T X1,Y1:H0=113:H1=210:XL=0:YL=1:XR=0:YR=
-1
20200 RETURN
29999 REM ------------TITEL------------
30000 GRAPHICS 17:POKE 77,0
30010 POSITION 3,5:? #6;"PANZERSCHLACHT"
:POSITION 0,14:? #6;"(C)1985 BY ASKI-WAR
E"
30020 POSITION 3,20:? #6;"DRUECKE  START
"
30030 IF PEEK(53279)<>6 THEN 30030
30040 RETURN
31999 REM --------ZEICHENSATZ--------
32000 P=PEEK(106)-8:Q=P*256:POKE 559,0:P
OKE 77,128:FOR I=0 TO 1023:POKE Q+I,PEEK
(57344+I):NEXT I
32005 RESTORE 32100
32010 READ N:IF N<>-1 THEN A=Q+N*8:FOR I
=A TO A+7:READ C:POKE I,C:NEXT I:GOTO 32
010
32020 RETURN
32100 DATA 3,254,254,254,254,254,254;254
,0
32110 DATA 56,130,68,40,16,40,68,130,0
32120 DATA 50,16,16,214,254,254,254,254,
198
32130 DATA 42,25,58,124,255,255,62,28,24
32140 DATA 43,252,252,120,127,120,252,25
2,0
32150 DATA 41,24,28,62,255,255,124,58,25
32160 DATA 49,198,254,254,254,254,214,16
,16
32170 DATA 45,24,56,124,255,255,62,92,15
2
32180 DATA 47,63,63,30,254,30,63,63,0
32190 DATA 46,152,92,62,255,255,124,56,2
4
32200 DATA 4,0,0,0,16,0,0,0,0
32300 DATA -1
```

Korrektur zum Apple-Programm „Music-Editor"
aus der Computronic 1/86

```
2455  FOR Z = NX - 2 TO NX + 2: HCOLOR= 0: HPLOT Z,NY TO Z,NY + 2: NEXT : HCOLOR= 3
3380  For I = NX - 2 TO NX + 2: HCOLOR= 0: HPLOT I,NY TO I,NY + 2: NEXT I HCOLOR= 3
```

Optimiertes Basic-Programm für den Profi

von Oliver Steinmeier

Dieser Artikel soll einige Tips geben, wie BASIC-Programme optimaler gestaltet werden. Diese Optimierung bewirkt dann auch normalerweise eine Erhöhung der Arbeitsgeschwindigkeit des Programms.

Der Autor geht von keinem speziellen Computer aus. Die hier abgedruckten Tips gelten für fast alle herkömmlichen BASIC-Homecomputer, da deren Interpreter normalerweise ähnlich arbeiten. Zeitangaben beziehen sich stets auf den Apple II, sind jedoch auf andere Rechner übertragbar.

Es wird von Basic-Programmierern immer wieder verlangt, strukturierte und damit lesbare Programme zu schreiben. Das bedeutet unter anderem, daß ein klarer Aufbau vorhanden sein muß. Am Anfang sollte das Hauptprogramm stehen, daß im Prinzip nur noch aus Unterprogrammaufrufen besteht.

Unter dem Gesichtspunkt der Geschwindigkeit-Problematik ist dies jedoch oft nicht mehr sinnvoll:

Wenn der Interpreter einen Verzweigungsbefehl verarbeitet, muß er zunächst das Programm nach dem Anfang der Zeile durchsuchen. Es ist natürlich nicht optimal, jedesmal vom Programmanfang an zu suchen.

Hat die Zeilennummer, zu der verzweigt werden soll, eine höhere Zeilennummer als die Zeile mit dem Sprungbefehl, so beginnt der Interpreter den Suchvorgang erst hinter dieser Programmzeile, da die gesuchte Zeile garantiert (auf Grund des Systems der aufeinanderfolgenden Zeilen) erst später folgt.

Hat die gesuchte Zeile jedoch eine kleinere Zeilennummer als die Sprungzeile, so bleibt dem Computer keine andere Wahl, er muß das Programm vom Anfang an untersuchen.

Eine Zeile wird also vom Interpreter dann besonders schnell gefunden, wenn sie möglichst nahe am Anfang des Programms steht, oder wenn sie möglichst bald nach der aufrufenden Zeile kommt.

Wenn das aufzurufende Unterprogramm häufig (z.B. in einer Schleife) angesprungen werden muß, so läßt sich auf dieser Art unter Umständen einige Zeit einsparen. Natürlich läßt sich das nicht immer mit dem gut strukturierten Programmierstil verbinden, aber in zeitkritischen Programmen sollte man dieser Methode dann trotzdem Vorrang gewähren.

Durch geschickte Verwendung von Variablen läßt sich ebenfalls einiges an Rechenzeit einsparen:

Es ist ein Vorurteil, daß einfache Integer-Variablen (sofern sie im Basic enthalten sind) einen Geschwindigkeitsvorteil bringen. Beim Apple und beim C64 werden Integer-Variablen mit dem '%'-Zeichen gekennzeichnet. Sie haben einen Rechenbereich von -32768 bis +32767 und können, wie ihr Name schon sagt, nur ganze Zahlen enthalten. Sie werden häufig eingesetzt, um Speicherplatz und Zeit zu sparen. Einfache Integer-Variablen (wie z. B. A%) sparen jedoch noch nicht einmal ein einziges Byte des kostbaren Speicherplatzes. Zwar benötigen sie zur Darstellung ihres Zahlenbereichs nur 16 Bit, also 2 Byte, (Fließkommavariablen benötigen normalerweise 5 Byte, um ihren Bereich abzudecken), die eingesparten 3 Byte werden jedoch mit Null-Bytes gefüllt, um mit ein und derselben Routine auf Integer- und Floating-Point-Variablen zugreifen zu können. Das liegt daran, daß beide Variablentypen normalerweise in einem Speicherbereich liegen. Verwendet man jedoch dimensionierte Integer-Vektoren (Arrays), so werden pro Wert wirklich nur 2 Byte gespeichert, denen dann eine eigene Routine zur Verfügung steht. Der Grund dafür, daß diese auch keine Rechenzeit einsparen, liegt darin, daß sie zu interen Rechnungen zunächst ins Fließkommaformat umgewandelt werden müssen. Die Integer-Vektoren sparen also keine Zeit ein, sondern können nur „langsam" verarbeitet werden, wie das nachfolgende Beispiel zeigt:

```
Programm 1:
10 FOR I=1 TO 2000
20 A%=A%+1
30 NEXT I
Zeit: ca. 9.25 Sek.
```

```
Programm II:
10 FOR I=1 TO 2000
20 A=A+1
30 NEXT I
Zeit: ca. 8.09 Sek.
```

Das „Fließkomma-Programm" benötigt mehr als eine Sekunde weniger als das entsprechende „Integer-Programm".

Weitere kostbare Rechenzeit (diesmal sogar extrem viel) kann man einsparen, wenn man häufig benutzte Zahlenkonstanten am Anfang eines Programms einer Variablen zuweist:

```
Programm I:
10 FOR I=1 TO 2000
20 A=3.25 E 17
30 NEXT I
Zeitdauer: ca. 22 Sekunden
```

```
Programm II:
 5 A=3.25 E 17
10 FOR I=1 TO 2000
20 A=B
30 NEXT I
Zeitdauer für dieses Programm:
ca. 5.1 Sekunden
```

Der Grund für die vierfache Geschwindigkeit des zweiten Programms liegt darin, daß bei Programm I die Konstante 3.25 E 17 zweitausend mal in eine Fließkommazahl umgewandelt werden muß, während sie bei Version II bereits nach Zeile 5 als Fließkommazahl in Variable B gespeichert ist.

Einige Variablen werden in Basic-Programmen besonders häufig benutzt. Normalerweise sind dies die Laufvariablen und Zähler für Schleifen. Wie die beiden folgenden Programmen zeigen, ist es durchaus sinnvoll, diese oft gebrauchten Variablen ganz am Anfang eines Programmes vor allen anderen Variablen einmal zu benutzen (z.B. auf Null setzen). Die Interpreter legen nämlich die Variablen in der Reihenfolge ihres Auftretens im Programm im Speicher ab. Wird eine angesprochen, so muß der gesamte Variablen-Speicherbereich nach der Variable durchsucht werden. Dann ist natürlich klar, daß häufig benutzte Variablen günstigerweise ganz am Anfang der Liste stehen und somit am Anfang des Programms einmal benutzt werden müssen.

```
Programm I:
10 A=O:B=O:C=O:D=O:E=O:F=O:
   G=O:H=O:O=O:P=O:Q=O
20 FOR I=1 TO 2000
30 I=I:J=A
40 NEXT I
Zeit ca. 10.24 Sekunden
```

```
Programm II:
 5 I=O:J=O
10 A=O:B=O:C=O:D=O:E=O:F=O:
   G=O:H=O:O=O:P=O:Q=O
20 FOR I=1 TO 2000
30 J=I:J=A
40 NEXT I
Zeit ca. 7.36 Sekunden
```

Wenn am Ende eines BASIC-Unterprogramms vor dem Rücksprung ein weiteres Unterprogramm aufgerufen werden muß,

Lesen Sie weiter auf Seite 85

SUPER! – Jetzt auch Checksummer für CPC 464

CHECKSUMMER 1.0 CPC, ist für den CPC 464/664 entwickelt worden.
Wie funktioniert unser Checksummer?
Tippen Sie zunächst das Programm in Ihren Computer ein. Bei der Eingabe ist zu beachten, daß auch alle Zwischenräume (Leerzeichen) mit dem Listing übereinstimmen. Die REM-Zeilen müssen ebenfalls mitübernommen werden.

Sichern Sie das Programm auf KASSETTE oder Diskette bevor Sie es starten!

Nachdem Sie es gespeichert haben, aktivieren Sie das Programm mit „RUN 65200“. Es erscheint die Meldung: CHECKSUMMER AKTIV FUNKTIONSTASTE 0. Dies bedeutet, wenn Sie die Taste 0 des Zehnerblocks betätigen, wird der Checksummer aufgerufen.
Speichern Sie das Programm jetzt mit dem Befehl SAVE “CHECKV1“,A auf Kassette oder Diskette ab. Nur so können Sie es später mit Merge an Ihre Programme anhängen.

Wie benutzt man den Checksummer?

Wenn Sie ein Programm mit CHECK überprüfen wollen, laden Sie zunächst das zu prüfende Programm (wenn Sie's nicht ohnehin schon im Speicher haben), dann geben Sie den Befehl MERGE “CHECKV1“ um das Prüfprogramm anzuhängen. Wichtig ist, daß das zu prüfende Programm keine Zeilennummer besitzt, die größer als 65199 ist. Jetzt aktivieren Sie den Checksummer mit RUN 65200. Nun kann jederzeit die Prüfung mit der Funktionstaste 0 aufgerufen werden. Vergleichen Sie jeweils die im Heft abgedruckten Prüfsummen mit denen, die Ihr Programm ausgibt. Falls Sie einmal einen Fehler feststellen, müssen Sie nicht das Programm neu laden, sondern können ihn normal berichtigen und dann wieder mit der Funktionstaste O das Check-Programm starten. Möchten Sie den Checksummer löschen, geben Sie einfach den Befehl DELETE 65200-ein.

Anmerkung zu Checksummer 1.0 CPC:
Groß-Kleinschreibung beachten!
Leerzeichen innerhalb einer Zeile müssen dem ORIGINAL-Listing entsprechen!
Leerzeichen am Ende einer Zeile werden ignoriert!
Umgewandelte TOOKEN werden automatisch in den Ursprungswert zurückgesetzt!
REM-Zeilen müssen eingegeben werden!

Unser Spezialist Frank Brall arbeitet zur Zeit an einem Checksummer für C-16/116. Also C-16-Fans achtet auf die nächste Computronic-Ausgabe!

ACHTUNG: Einige Listings enthalten das Zeichen ”←“, dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben). – Das Zeichen ” ∧ “ steht für den Pfeil nach oben. Das Zeichen ”\“ steht für das (engl.) Pfund-Symbol. (Taste neben Restore).

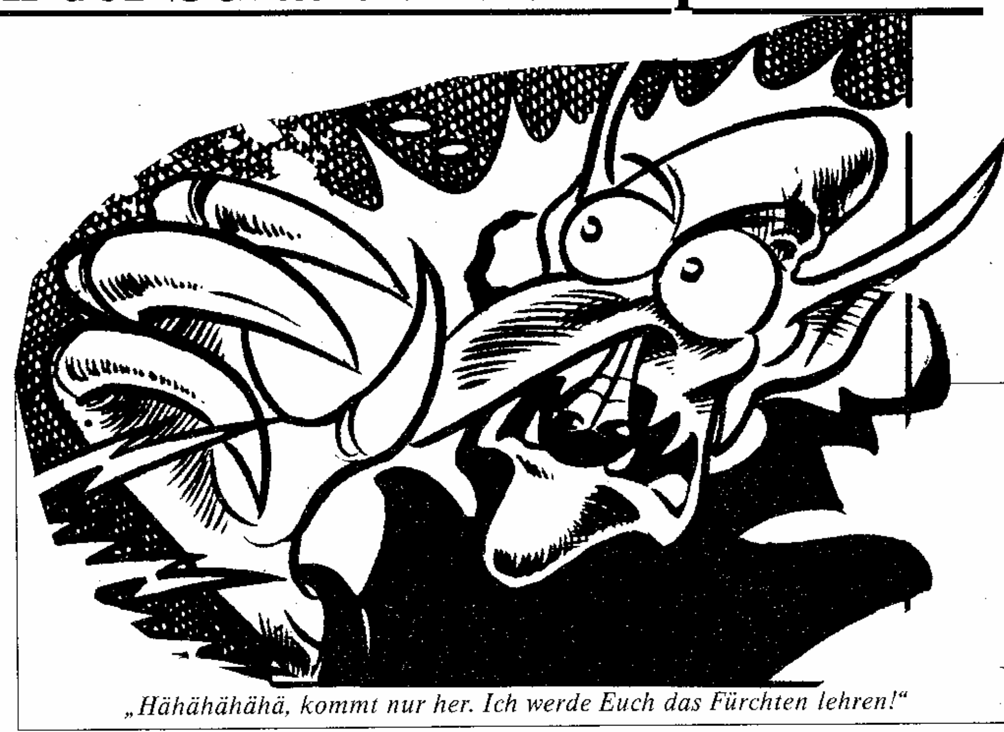
```
65200 '*** Tronic CPC Checksummer ***
65210 '*** Autor Frank Brall 1985 ***
65220 '          VERSION 1.0
65230 KEY 128,"65530 REM TRONIC"+CHR$(13
)+"run 65250"+CHR$(13)
65240 PRINT:PRINT"CHECKSUMMER AKTIV  FUN
KTIONSTASTE 0":PRINT:END
65250 CLS:PRINT"Tronic-Verlag Checksumme
r Version 1.0"
65260 PRINT:PRINT:PRINT:PRINT"<B>ILDSCHI
RM  <D>RUCKER"
65270 E$=INKEY$:E$=LOWER$(E$):IF E$="" T
HEN 65270
65280 IF E$="b" THEN a=0:GOTO 65310
65290 IF E$="d" THEN a=8:GOTO 65310
65300 GOTO 65200
65310 ADRESS=368:GCHECK=0:CLS:PRINT#A,"C
HECK V1:PRUEFSUMMEN:":PRINT#A:PRINT#A,"Z
EILENNR.:  SUMMEN"
65320 LAENGE=(PEEK(ADRESS+1)*256)+PEEK(A
DRESS)
65330 ZNUM=(PEEK(ADRESS+3)*256)+PEEK(ADR
ESS+2)
65340 IF ZNUM=65200 THEN IF F=0 THEN 654
80 ELSE PRINT #A, USING "#####";ZNUM1;:P
RINT#A,"-      ";:PRINT#A,":";:PRINT #A,
USING "######";ZCHECK:F=0:GOTO 65480
65350 IF ZNUM=65500 THEN 65510
65360 ZCHECK=0:FOR I=0 TO LAENGE-6
65370 WERT=PEEK(I+ADRESS+4)
65380 IF WERT =32 THEN 65410
65390 ZCHECK=ZCHECK+WERT+(I*WERT)
65400 IF ZCHECK>65535 THEN ZCHECK=ZCHECK
-65536:GOTO 65400
65410 NEXT I
65420 IF F=0 THEN ZCHECK1=ZCHECK:ZNUM1=Z
NUM:ADRESS=ADRESS+LAENGE:F=1:GOTO 65320
65430 F=0:ZNUM2=ZNUM
65440 PRINT #A,USING "#####";ZNUM1;:PRIN
T#A,"-";:PRINT #A,USING "#####";ZNUM2;:P
RINT#A,":";:PRINT #A,USING "######";ZCHE
CK+ZCHECK1
65450 GCHECK=GCHECK+ZCHECK+ZCHECK1
65460 IF GCHECK>65535 THEN GCHECK=GCHECK
-65536:GOTO 65460
65470 ADRESS=ADRESS+LAENGE:GOTO 65320
65480 PRINT#A, "-----------------------
-"
65490 PRINT #a,"GESAMTCHECKSUMME: "GCHEC
K
65500 GCHECK=0:A=0:PRINT#A:PRINT#A,"CHEC
KSUMMER SELBSTTEST !":GOTO 65360
65510 IF GCHECK=62078 THEN PRINT#A,"--EN
DE--":END
65520 PRINT "FEHLER IM CHECKSUMMER !":SO
UND 1,100,500
65530 REM TRONIC
```

„Fix und fertig“ ist der Checksummer 1.0 CPC natürlich auch erhältlich. Einfach anfordern unter:
Bestellnummer: SRV 10K (Kassette) . . . . 10 DM   SRV 10D (Diskette) . . . . 20 DM

„Midnight“ *Geschicklichkeitsspiel für den Schneider-Computer*

Auf der Jagd nach der Schatztruhe des Spukhauses!

Ziel des Spiels ist es, die im Spukhaus deponierte Truhe zu erreichen. Dabei müssen die Euch feindlich gesinnten Geister und die Feuerkugel übersprungen werden, weil jede Berührung mit ihnen ein Leben, von denen Ihr insgesamt drei besitzt, kostet. Die Spieße in der 1. und 3. Etage müssen ebenfalls übersprungen werden. Um die Truhe auch öffnen zu können, müßt Ihr vorher im 3. Stock durch einen Sprung einen unsichtbaren Lichtschalter berühren. Danach wird im 2. Stock ein Schlüssel sichtbar, der ebenfalls per Sprung erreichbar ist. Im 5. Stockwerk versperrt Euch der dortige Etagengeist den Zugang zur Truhe. Erst wenn der Geist mit Hilfe eines Pfahls aus dem Weg geräumt ist, kann die Truhe in Euren Besitz gelangen. Es wird gegen die Uhr gespielt, d. h. wenn der Mond von links nach rechts seinen Weg beendet hat, wird ein Leben abgezogen. Ihr besitzt allerdings auch die Möglichkeit das Haus vorher zu verlassen, so daß zwar Punkte, aber keine Leben verloren gehen. (S.G.)



„Hähähähähä, kommt nur her. Ich werde Euch das Fürchten lehren!“

Achtet darauf, daß Ihr auf dem Weg zur Schatztruhe nicht Euer kostbares Leben verliert!

Steuerung: Gespielt wird mit Joystick, gesprungen wird durch Drükken des Feuerknopfes und Halten des Joysticks in Sprungrichtung.

MIDNIGHT-Vorprogramm

G E I S T E R

```
5 REM ---zeichen---
10 SYMBOL AFTER 129
15 SYMBOL 129,0,24,28,24,16,0,0,0
20 SYMBOL 130,0,0,0,0,0,56,56,92
25 SYMBOL 131,94,90,0,0,0,0,0,0
30 SYMBOL 132,0,0,24,24,36,36,68,70
35 SYMBOL 133,0,0,24,24,24,24,40,44
40 SYMBOL 134,0,24,56,24,8,0,0,0
45 SYMBOL 135,0,0,0,0,0,28,28,58
50 SYMBOL 136,122,90,0,0,0,0,0,0
55 SYMBOL 137,0,0,24,24,36,36,34,98
60 SYMBOL 138,0,0,24,24,24,24,20,52
65 SYMBOL 139,0,24,24,24,0,0,0,0
70 SYMBOL 140,0,0,0,0,0,60,60,90
75 SYMBOL 141,90,90,0,0,0,0,0,0
80 SYMBOL 142,0,0,24,24,36,36,36,36
85 SYMBOL 143,0,0,0,0,24,28,24,16
90 SYMBOL 144,26,30,28,24,24,0,0
95 SYMBOL 145,0,0,0,0,6,30,26,3
100 SYMBOL 146,0,0,0,0,24,56,24,8
105 SYMBOL 147,88,120,56,24,0,0,0
110 SYMBOL 148,0,0,0,0,96,120,88,192
115 SYMBOL 149,56,84,168,124,50,48,96,64
120 SYMBOL 150,0,0,0,96,34,226,97,225
125 SYMBOL 151,49,61,190,126,54,34,34,68
130 SYMBOL 152,2,0,16,4,0,32,8,72
135 SYMBOL 153,26,56,60,52,36,36,36,0
140 SYMBOL 154,0,0,0,8,24,24,24,24
145 SYMBOL 154,0,0,0,8,24,24,24,24
150 SYMBOL 155,0,24,20,60,26,52,28,8
155 SYMBOL 156,0,0,16,16,16,24,24,24
160 SYMBOL 157,0,127,65,127,119,127,127,
127
165 SYMBOL 158,0,0,0,0,0,24,24,0
170 SYMBOL 159,0,0,0,24,24,8,8,24
175 SYMBOL 160,0,0,0,0,96,48,24,12
180 SYMBOL 161,255,129,255,129,255,129,2
55,129
185 SYMBOL 162,0,0,0,24,28,12,24,28
190 SYMBOL 163,191,253,239,0,0,0,0,0
195 SYMBOL 164,7,3,7,6,7,7,3,7
200 SYMBOL 165,224,96,224,192,224,224,96
,224
205 SYMBOL 166,1,3,4,12,16,48,64,192
210 SYMBOL 167,128,192,32,48,8,12,2,3
215 SYMBOL 168,0,0,0,0,0,0,85,170
220 SYMBOL 169,123,45,31,13,7,3,1,0
225 SYMBOL 170,255,182,251,95,255,253,23
9,255
```

S T U N D E !

IST ANGESAGT

```
230 SYMBOL 171,0,44,60,126,126,60,52,16
235 SYMBOL 172,24,60,60,126,126,255,255,
255
240 SYMBOL 173,255,255,255,126,126,60,60
,24
245 SYMBOL 174,0,0,0,0,0,20,62,255
250 SYMBOL 175,1,3,15,15,29,27,63,63
255 SYMBOL 176,127,127,55,127,123,47,63,
127
260 SYMBOL 177,63,61,31,11,15,7,3,3
265 SYMBOL 178,128,192,240,240,184,216,2
52,252
270 SYMBOL 179,254,190,236,254,222,244,2
52,254
275 SYMBOL 180,252,188,248,208,240,224,1
92,192
280 SYMBOL 181,243,243,242,246,246,254,1
26,126
285 SYMBOL 182,126,126,126,62,126,126,12
4,126
290 SYMBOL 183,7,3,3,3,3,3,3,3
295 REM ---bild---
300 INK 0,10:INK 1,0:INK 2,17:INK 3,19:M
ODE 1:BORDER 10
305 a$="Midnight":x=110:y=200:xr=5:yr=10
:p=1:v=-2:GOSUB 325
310 LOCATE 5,14:PEN 2:PRINT"von Martin W
olfgramm (c)1985"
315 LOCATE 4,24:PEN 3:PRINT "Bitte warte
n:Loading...":PRINT:PRINT
320 RUN "!Midnight1"
325 PEN p:LOCATE 1,1:PRINT a$
330 FOR n=384 TO 400 STEP 2
335 FOR i=1 TO LEN(a$)*16 STEP 2
340 IF TEST(i,n)<>p THEN 350
345 FOR r=0 TO xr-1:PLOT r+x,y,p:DRAWR 0
,yr,p:NEXT
350 x=x+xr:NEXT i
355 x=x-LEN(a$)*8*xr+v:y=y+yr:NEXT
360 RETURN
```

```
CHECK V1:PRUEFSUMMEN:

ZEILENNR.:  SUMMEN
    5-   10: 10758
   15-   20: 20743
   25-   30: 22794
   35-   40: 21186
   45-   50: 18852
   55-   60: 25636
   65-   70: 20228
   75-   80: 21226
   85-   90: 18376
   95-  100: 17792
  105-  110: 27640
  115-  120: 43209
  125-  130: 29180
  135-  140: 23791
  145-  150: 22327
  155-  160: 35170
  165-  170: 17136
  175-  180: 46666
  185-  190: 25325
  195-  200: 43296
  205-  210: 28603
  215-  220: 22982
  225-  230: 59019
  235-  240: 60091
  245-  250: 28116
  255-  260: 34136
  265-  270: 84692
  275-  280: 77533
  285-  290: 33362
  295-  300: 28737
  305-  310: 59716
  315-  320: 17729
  325-  330: 25349
  335-  340: 51643
  345-  350: 71445
  355-  360: 49549
--------------------------------
GESAMTCHECKSUMME:  64385
```

MIDNIGHT-Hauptprogramm

```
5  REM ********************************
10 REM *                              *
15 REM *       M I D N I G H T        *
20 REM *                              *
25 REM *        (c) 1985 von          *
30 REM *                              *
35 REM *      Martin Wolfgramm        *
40 REM *                              *
45 REM ********************************
50 CLS
55 REM ---zwischenbild---
60 MODE 1:BORDER 10:INK 0,10:INK 1,6:INK
 2,0:INK 3,19:PAPER 0
65 hisc=PEEK(30000)*256+PEEK(30001)
70 LOCATE 17,4:PEN 1:PRINT "Midnight"
75 LOCATE 16,5:PRINT "----------"
80 LOCATE 4,7:PEN 2:PRINT "(c) 1985    v
on Martin Wolfgramm"
85 LOCATE 4,9:PRINT "Hoechstzahl:";hisc
90 LOCATE 4,11:PRINT"Punkte      :";sc
95 LOCATE 4,15:PRINT"Steuerung   : Joysti
ck"
100 LOCATE 4,17:PRINT"Sprung      : Feuer
taste"
105 LOCATE 4,19:PRINT"Ende        : Break
-Taste"
110 LOCATE 4,24:PEN 3:PRINT "Druecke Tas
te"
115 FOR i=1 TO 2:IF INKEY(66)=0 THEN GOT
O 1630:NEXT i
120 CALL &BB06
125 CLEAR
130 REM ---bildschirmaufbau---
135 MODE 0:INK 0,0:BORDER 0:PAPER 0:INK
1,16:INK 2,6:INK 3,11:INK 4,23:INK 5,25:
INK 6,24:INK 7,3:INK 8,17:INK 9,15:INK 1
0,7:INK 11,10:INK 12,9:INK 13,18:INK 14,
8:INK 15,26
140 WINDOW #1,1,20,25,25
145 hisc=PEEK(30000)*256+PEEK(30001):sc=
0:liv=3:RANDOMIZE TIME
150 nx=3:nyo=22:nyu=23:gy(1)=23:gy(2)=20
:gy(3)=17:gy(4)=11:fy=14:f=16:p=0:z=2:g=
0:s=0:zz=4:gn(1)=16:gn(2)=12:gn(3)=10:gn
(4)=14:q=0:o=0:sp=0:ver=0:ps=0
155 DIM b(20,25),m(20,25),g(20,25)
160 ON BREAK GOSUB 1630:CLS:PRINT CHR$(2
2);CHR$(1)
165 LOCATE 2,13:PEN 12:PRINT CHR$(174)
170 PRINT CHR$(175);CHR$(170);CHR$(178)
175 PRINT CHR$(176);CHR$(170);CHR$(179)
180 PRINT CHR$(176);CHR$(170);CHR$(179)
185 PRINT CHR$(177);CHR$(170);CHR$(180)
190 LOCATE 2,18:PEN 7:PRINT CHR$(181)
195 FOR i=18 TO 21:LOCATE 2,i:PRINT CHR$
(182):NEXT i
200 FOR i=1 TO 4:LOCATE i,23:PRINT CHR$(
170):NEXT i
205 FOR i=1 TO 4:PEN 13:LOCATE i,22:PRIN
T CHR$(170):NEXT i
210 FOR i=1 TO 20:LOCATE i,24:PRINT CHR$
(170):NEXT i
215 LOCATE 20,23:PRINT CHR$(170):LOCATE
20,22:PRINT CHR$(170)
220 j=12:FOR i=5 TO 9:j=j-1:PEN 2:LOCATE
 i,j:PRINT CHR$(166):NEXT i
225 FOR i=10 TO 14:LOCATE i,6:PRINT CHR$
(168):NEXT i
230 j=6:FOR i=15 TO 19:j=j+1:LOCATE i,j:
PRINT CHR$(167):NEXT i
235 LOCATE 16,8:PEN 11:PRINT CHR$(169)
240 LOCATE 16,7:PRINT CHR$(170):LOCATE 1
6,6:PRINT CHR$(170)
245 PEN 15:LOCATE 16,5:PRINT CHR$(171):L
```



```
OCATE 17,4:PRINT CHR$(171)
250 FOR ii=12 TO 23:PEN 11:LOCATE 20,ii:
PRINT CHR$(165):b(20,ii)=165:NEXT ii
255 FOR i=12 TO 21:LOCATE 4,i:PRINT CHR$
(164):b(4,i)=164:NEXT i
260 LOCATE 4,22:PEN 9:PRINT CHR$(183)
265 LOCATE 4,23:PRINT CHR$(183)
270 FOR i=8 TO 16:PEN 11:LOCATE i,9:PRIN
T CHR$(163):b(i,9)=163:NEXT i
275 FOR j=12 TO 21 STEP 3
280 FOR i=5 TO 19:LOCATE i,j:PRINT CHR$(
163):b(i,j)=163:NEXT i
285 NEXT j
290 RESTORE 305:PRINT CHR$(22);CHR$(0):F
OR a=1 TO 12:READ i,j
295 LOCATE i,j:PRINT CHR$(161):b(i,j)=16
1:NEXT a
300 PRINT CHR$(22);CHR$(1):GOTO 310
305 DATA 17,12,17,13,17,14,5,15,5,16,5,1
7,5,18,5,19,5,20,19,21,19,22,19,23,0,0,0
,0,0,0,0,0
310 RESTORE 325:FOR a=1 TO 5:READ i,j
315 LOCATE i,j:PRINT CHR$(162):b(i,j)=16
2:NEXT a
320 GOTO 330
325 DATA 12,9,10,12,14,15,15,18,9,21
330 LOCATE 7,11:PEN 7:PRINT CHR$(157):b(
7,11)=157
335 LOCATE 12,17:PEN 8:PRINT CHR$(156):b
(12,17)=156
340 LOCATE 11,23:PRINT CHR$(156):b(11,23
)=156
345 RESTORE 360:FOR a=1 TO 5:READ i,j
350 LOCATE i,j:PEN 9:PRINT CHR$(158):b(i
,j)=158:NEXT a
355 GOTO 365
360 DATA 12,12,8,15,8,18,17,18,7,21
365 i=INT(RND*9)+8:IF i=12 THEN GOTO 365
370 LOCATE i,9:PEN 7:PRINT CHR$(160):b(i
,9)=160
375 a=RND
380 IF a>0.5 THEN b(7,18)=1 ELSE b(8,18)
=1
385 a=RND
390 IF a>0.5 THEN b(12,21)=1 ELSE b(13,2
1)=1
395 i=INT(RND*14)+6:IF i=14 OR i=8 THEN
GOTO 395
400 b(i,15)=2
405 LOCATE 18,13:PEN 10:PRINT CHR$(150)
410 LOCATE 18,14:PRINT CHR$(151):b(18,14
)=165
415 LOCATE 19,13:PEN 9:PRINT CHR$(152)
420 LOCATE 19,14:PRINT CHR$(153):LOCATE
19,14:PEN 6:PRINT CHR$(154)
425 RESTORE 435:FOR a=1 TO 4:READ i,j
430 b(i,j)=3:NEXT a:GOTO 440
435 DATA 17,11,6,17,5,14,19,20
440 b(18,11)=165:b(6,11)=164:b(17,9)=162
:b(7,9)=162
445 LOCATE #1,2,25:PEN #1,14:PRINT #1,"L
EBEN : ";liv
450 REM ---hauptprogramm---
455 LOCATE z,2::PEN 15:PRINT CHR$(172):L
OCATE z,3:PRINT CHR$(173):e=133
460 LOCATE nx,nyo:PEN 1:PRINT CHR$(129)
465 LOCATE nx,nyo:PEN 2:PRINT CHR$(130)
470 LOCATE nx,nyu:PEN 2:PRINT CHR$(131)
475 LOCATE nx,nyu:PEN 3:PRINT CHR$(e):SO
UND 1,2000,2,5,,,2:ax=nx:ayo=nyo:ayu=nyu
480 IF q=1 THEN RETURN
485 FOR a=1 TO 750:NEXT a
490 PRINT CHR$(22);CHR$(0)
495 LOCATE 4,22:PEN 13:PRINT CHR$(170)
500 LOCATE 4,23:PEN 7:PRINT CHR$(170)
505 nx=nx+1
510 q=1:LOCATE ax,ayo:PEN 13:PRINT CHR$(
170)
515 LOCATE ax,ayu:PEN 7:PRINT CHR$(170)
520 PRINT CHR$(22);CHR$(1):e=132:GOSUB 4
60
525 FOR a=1 TO 750:NEXT a
530 q=1:PRINT CHR$(22);CHR$(0):nx=nx+1
535 LOCATE ax,ayo:PEN 13:PRINT CHR$(170)
540 LOCATE ax,ayu:PEN 7:PRINT CHR$(170):
b(ax,ayu)=4
545 PRINT CHR$(22);CHR$(1):e=133:GOSUB 4
60:q=0
550 ON BREAK GOSUB  1630
555 xa=5:ayo=22:ayu=23:xn=5:nyo=22:nyu=2
3
560 gn(1)=16:gy(1)=23:gn(2)=12:gy(2)=20:
gn(3)=10:gy(3)=17:gn(4)=14:gy(4)=11:ga(1
)=16:ga(2)=12:ga(3)=10:ga(4)=14:f=16:fa=
16:fy=14
565 REM ---joystick---
570 ON BREAK GOSUB 1630
575 po=po+1:IF po=40 THEN po=0:GOSUB 159
5
580 IF JOY(0)=4 AND b(xn-1,nyu)<>164 THE
N GOTO 635
585 IF JOY(0)=8 AND b(xn+1,nyu)<>165 THE
N GOTO 710
590 IF JOY(0)=1 AND b(xn,nyu)=161 THEN n
yo=nyo-1:nyu=nyu-1:GOTO 790
595 IF JOY(0)=2 AND b(xn,nyu+1)=161 THEN
 nyo=nyo+1:nyu=nyu+1:GOTO 790
600 IF JOY(0)=20 AND b(xn,nyo-1)<>162 TH
EN GOTO 860
605 IF JOY(0)=24 AND b(xn,nyo-1)<>162 TH
EN GOTO 1010
610 FOR a=1 TO 400:NEXT a
615 ps=ps+1:IF ps=3 THEN ps=0:GOSUB 1300
620 GOSUB 1365
625 sp=sp+1:IF sp=19 THEN sp=0:GOSUB 125
5
630 GOTO 570
635 REM ---links gehen---
640 IF b(nx-1,nyo)=163 OR b(nx-1,nyu)=16
3 THEN GOTO 570
645 DI:PRINT CHR$(22);CHR$(0):LOCATE xa,
ayo
650 DI:IF b(xa,ayo)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " "
655 LOCATE xa,ayu
660 DI:IF b(xa,ayu)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " ":PRIN
T CHR$(22);CHR$(1)
665 m(xa,ayu)=0
670 xn=xn-1:xa=xn:ayo=nyo:ayu=ayu
```



```
675 DI:LOCATE xn,nyo:PEN 1:PRINT CHR$(13
4)
680 LOCATE xn,nyo:PEN 2:PRINT CHR$(135)
685 LOCATE xn,nyu:PEN 2:PRINT CHR$(136)
690 IF e=137 THEN e=138 ELSE e=137
695 LOCATE xn,nyu:PEN 3:PRINT CHR$(e)
700 SOUND 1,2000,2,5,,,2
705 EI:GOSUB 1205:GOTO 615
710 REM ---rechts gehen---
715 IF b(xn+1,nyo)=163 OR b(xn+1,nyu)=16
3 THEN GOTO 570
720 DI:PRINT CHR$(22);CHR$(0):LOCATE xa,
ayo
725 DI:IF b(xa,ayo)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " "
730 LOCATE xa,ayu
735 DI:IF b(xa,ayu)=157 THEN PEN 7:PRINT
 CHR$(157)
740 DI:IF b(xa,ayu)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " "
745 m(xa,ayu)=0:PRINT CHR$(22);CHR$(1)
750 xn=xn+1:xa=xn:ayo=nyo:ayu=nyu
755 DI:LOCATE xn,nyo:PEN 1:PRINT CHR$(12
9)
760 LOCATE xn,nyo:PEN 2:PRINT CHR$(130)
765 LOCATE xn,nyu:PEN 2:PRINT CHR$(131)
770 IF e=132 THEN e=133 ELSE e=132
775 LOCATE xn,nyu:PEN 3:PRINT CHR$(e)
780 SOUND 1,2000,2,5,,,2
785 m(xn,nyu)=1:EI:GOSUB 1205:GOTO 615
790 REM ---oben/unten gehen---
795 DI:PRINT CHR$(22);CHR$(0):LOCATE xa,
ayo
800 DI:IF b(xa,ayo)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " "
805 LOCATE xa,ayu
810 DI:IF b(xa,ayu)=161 THEN PEN 11:PRIN
T CHR$(161) ELSE DI:PEN 0:PRINT " ":m(xa
,ayu)=0
815 PRINT CHR$(22);CHR$(1)
820 xa=xn:ayo=nyo:ayu=nyu
825 DI:LOCATE xn,nyo:PEN 1:PRINT CHR$(13
9)
830 LOCATE xn,nyo:PEN 2:PRINT CHR$(140)
835 LOCATE xn,nyu:PEN 2:PRINT CHR$(141)
840 LOCATE xn,nyu:PEN 3:PRINT CHR$(142)
845 SOUND 1,2000,2,5,,,2
850 m(xn,nyu)=1:EI:GOSUB 1205:IF o=1 THE
N RETURN
855 GOTO 615
860 REM ---links springen---
865 IF b(xn,nyo-1)=161 THEN GOTO 570
870 DI:PRINT CHR$(22);CHR$(0):LOCATE xa,
ayo:PEN 0:PRINT " "
875 LOCATE xa,ayu:PEN 0:PRINT " "
880 m(xa,ayu)=0
885 xn=xn:nyo=nyo-1:nyu=nyu-1
890 PRINT CHR$(22);CHR$(1)
895 ENV 5,5,2,6
900 SOUND 1,120,0,2,5,,5
905 LOCATE xn,nyo:PEN 1:PRINT CHR$(146)
910 LOCATE xn,nyu:PEN 2:PRINT CHR$(147)
915 LOCATE xn,nyu:PEN 3:PRINT CHR$(148):
GOSUB 1560
920 FOR i=1 TO 2:DI:IF b(xn-1,nyo)=162 T
HEN GOTO 980
925 IF b(xn-1,nyo)=164 THEN 980
930 IF b(xn-1,nyo)=161 THEN 980
935 xa=xn:ayo=nyo:ayu=nyu
940 PRINT CHR$(22);CHR$(0)
945 LOCATE xa,ayo:PEN 11:PRINT CHR$(163)
950 LOCATE xa,ayu:PEN 0:PRINT " "
955 PRINT CHR$(22);CHR$(1):xn=xn-1
960 LOCATE xn,nyo:PEN 1:PRINT CHR$(146)
965 LOCATE xn,nyu:PEN 2:PRINT CHR$(147)
970 LOCATE xn,nyu:PEN 3:PRINT CHR$(148)
975 GOSUB 1560:NEXT i
980 DI:PRINT CHR$(22);CHR$(0)
985 xa=xn:ayo=nyo:ayu=nyu
990 DI:LOCATE xa,ayo:PEN 11:PRINT CHR$(1
63)
995 DI:LOCATE xa,ayu:PEN 0:PRINT " "
1000 PRINT CHR$(22);CHR$(1)
1005 xa=xa:ayo=ayo+1:ayu=ayu+1:xn=xn:nyo
=nyo+1:nyu=nyu+1:GOTO 675
1010 REM ---rechts springen---
1015 IF b(xn,nyo-1)=161 THEN GOTO 570
1020 DI:PRINT CHR$(22);CHR$(0):LOCATE xa
,ayo:PEN 0:PRINT " "
1025 LOCATE xa,ayu:PEN 0:PRINT " "
1030 m(xa,ayu)=0
1035 xn=xn:nyo=nyo-1:nyu=nyu-1
1040 PRINT CHR$(22);CHR$(1)
1045 ENV 5,5,2,6
1050 SOUND 1,120,0,2,5,,5
1055 LOCATE xn,nyo:PEN 1:PRINT CHR$(143)
1060 LOCATE xn,nyu:PEN 2:PRINT CHR$(144)
1065 LOCATE xn,nyu:PEN 3:PRINT CHR$(145)
:GOSUB 1560
1070 FOR i=1 TO 2:DI:IF b(xn+1,nyo)=162
THEN GOTO 1130
1075 IF b(xn+1,nyo)=165 THEN 1130
1080 IF b(xn+1,nyo)=161 THEN 1130
1085 xa=xn:ayo=nyo:ayu=nyu
1090 PRINT CHR$(22);CHR$(0)
1095 LOCATE xa,ayo:PEN 11:PRINT CHR$(163
)
1100 LOCATE xa,ayu:PEN 0:PRINT " "
1105 PRINT CHR$(22);CHR$(1):xn=xn+1
1110 LOCATE xn,nyo:PEN 1:PRINT CHR$(143)
1115 LOCATE xn,nyu:PEN 2:PRINT CHR$(144)
1120 LOCATE xn,nyu:PEN 3:PRINT CHR$(145)
1125 GOSUB 1560:NEXT i
1130 DI:PRINT CHR$(22);CHR$(0)
1135 xa=xn:ayo=nyo:ayu=nyu
1140 LOCATE xa,ayo:PEN 11:PRINT CHR$(163
)
1145 LOCATE xa,ayu:PEN 0:PRINT " "
1150 PRINT CHR$(22);CHR$(1)
1155 xa=xa:ayo=ayo+1:ayu=ayu+1:xn=xn:nyo
=nyo+1:nyu=nyu+1:GOTO 755
1160 REM ---fallen---
1165 ENV 2,5,2,1,5,-3,2
1170 SOUND 3,250,0,4,2,,10
1175 FOR i=1 TO 3
1180 nyo=nyo+1:nyu=nyu+1:o=1:GOSUB 790
1185 NEXT i:o=0
1190 PRINT CHR$(22);CHR$(0)
1195 LOCATE xn,nyo-1:PEN 11:PRINT CHR$(1
63):b(xn,nyo-1)=1
```



```
1200 GOTO 570
1205 REM ---ueberpruefung 1---
1210 IF g(xn,nyu)=10 AND p=2 THEN p=3:sc
=sc+500:PEN #1,14:LOCATE #1,2,25:PRINT #
1,"Punkte: ";sc;:zz=3:g(xn,nyu)=0
1215 IF b(xn,nyu)=3 THEN sc=sc+40:PEN #1
,14:LOCATE #1,2,25:PRINT #1,"Punkte: ";s
c;:b(xn,nyu)=0
1220 IF g(xn,nyu)=149 THEN GOTO 1455
1225 IF b(xn,nyu)=156 THEN GOTO 1455
1230 IF b(xn,nyu)=4 THEN ver=1:sc=sc-50:
PEN #1,14:LOCATE #1,2,25:PRINT #1,"Punkt
e: ";sc;:GOTO 1455
1235 IF b(xn,nyu)=157 AND p=3 THEN sc=sc
+500:PEN #1,14:LOCATE #1,2,25:PRINT #1,"
Punkte: ";sc;:GOTO 1415
1240 IF b(xn,nyu+1)=1 THEN GOTO 1160
1245 IF g(xn,nyu)=10 THEN GOTO 1455
1250 RETURN
1255 REM ---mond---
1260 DI:PRINT CHR$(22);CHR$(0)
1265 LOCATE z-1,2:PEN 0:PRINT " "
1270 LOCATE z-1,3:PEN 0:PRINT " "
1275 LOCATE z,2:PEN 15:PRINT CHR$(172)
1280 LOCATE z,3:PEN 15:PRINT CHR$(173)
1285 IF z=18 THEN INK 0,1
1290 IF z=19 THEN GOTO 1455
```

```
:PEN 0:PRINT " "
1385 PRINT CHR$(22);CHR$(1):LOCATE f,fy:
PEN 9:PRINT CHR$(155):g(fa,fy)=0:g(f,fy)
=149
1390 ENV 3,6,1,8
1395 ENT 3,10,-2,2
1400 SOUND 4,284,0,2,3,3,2
1405 IF m(f,fy)=1 THEN GOTO 1455
1410 EI:RETURN
1415 REM ---gewonnen---
1420 ENT 1,5,1,1,10,-1,1,10,1,1,5,-1,1
1425 RESTORE 1445
1430 READ a,b:IF a=-1 THEN GOTO 1450
1435 SOUND 7,a,b,7,,1
1440 GOTO 1430
1445 DATA 478,20,0,10,478,20,426,40,506,
45,478,300,-1,-1
1450 INK 0,0:ERASE b,m,g:GOTO 150
1455 REM ---verloren---
1460 r1=0:r2=0:r3=0
1465 ENT 1,5,1,1,10,-1,1,10,1,1,5,-1,1
1470 RESTORE 1490
1475 READ a,b:IF a=-1 THEN GOTO 1495
1480 SOUND 7,a,b,5,,1
1485 GOTO 1475
1490 DATA 319,50,402,40,426,30,402,30,47
8,300,-1,-1
```



MIDNIGHT




```
1295 z=z+1:PRINT CHR$(22);CHR$(1):EI:RET
URN
1300 REM ---geister---
1305 FOR i=1 TO zz
1310 j=INT(RND*3)-1
1315 ga(i)=gn(i)
1320 gn(i)=gn(i)+j:IF b(gn(i),gy(i))<>0
AND b(gn(i),ga(i))<>156 THEN gn(i)=ga(i)
:GOTO 1360
1325 g(ga(i),gy(i))=0:PRINT CHR$(22);CHR
$(0)
1330 IF b(ga(i),gy(i))=156 THEN PEN 8:LO
CATE ga(i),gy(i):PRINT CHR$(156);:GOTO 1
340
1335 PEN 0:LOCATE ga(i),gy(i):PRINT " "
1340 PRINT CHR$(22);CHR$(1)
1345 PEN 4:IF i=4 THEN PEN 5
1350 g(gn(i),gy(i))=149:IF i=4 THEN g(gn
(i),gy(i))=10
1355 LOCATE gn(i),gy(i):PRINT CHR$(149);
:IF m(gn(i),gy(i))=1 THEN GOTO 1455
1360 NEXT i:RETURN
1365 REM ---feuerkugel---
1370 fa=f:DI
1375 f=f-1:IF b(f,fy)<>0 THEN PRINT CHR$
(22);CHR$(0):LOCATE fa,fy:PEN 0:PRINT "
":g(fa,fy)=0::f=16:PRINT CHR$(22);CHR$(1
):EI:RETURN
1380 PRINT CHR$(22);CHR$(0):LOCATE fa,fy
```

```
1495 INK 0,0:IF ver=1 THEN liv=liv ELSE
liv=liv-1
1500 ver=0
1505 IF liv=0 THEN GOTO 1515
1510 ERASE b,m,g:GOTO 150
1515 PEN 11:PAPER 15:PRINT CHR$(22);CHR$
(0)
1520 LOCATE 5,14:PRINT "              "
1525 LOCATE 5,15:PRINT " GAME  OVER "
1530 LOCATE 5,16:PRINT "              "
1535 FOR a=1 TO 2000:NEXT a
1540 IF sc>hisc THEN hisc=sc
1545 hi1=(hisc MO
D 256):hi2=INT(hisc/256)
1550 POKE 30000,hi2:POKE 30001,hi1
1555 CALL &BB03:GOTO 60
1560 REM ---ueberpruefung 2---
1565 IF b(xn,nyo)=158 THEN ENT 2,100,-2,
4:SOUND 2,190,10,5,,2:g=g+1:b(xn,nyo)=16
3:sc=sc+25:LOCATE #1,2,25:PEN #1,14:PRIN
T #1,"Punkte: ";sc;
1570 IF g=5 THEN sc=sc+100:LOCATE #1,7,2
5:PEN #1,14:PRINT #1,"Punkte: ";sc;
1575 IF b(xn,nyo)=2 THEN SOUND 2,478,10,
2:b(xn,nyo)=163:s=1:sc=sc+30:PEN #1,14:L
OCATE #1,2,25:PRINT #1,"Punkte: ";sc;:LO
CATE 18,18:PEN 6:PRINT CHR$(159):b(18,18
)=159
1580 IF b(xn,nyo)=159 AND s=1 THEN ENV 4
```



```
,50,2,10:SOUND 3,65,80,2,4:p=p+1:b(xn,ny
o)=163:sc=sc+75:LOCATE #1,2,25:PEN #1,14
:PRINT #1,"Punkte: ";sc;
1585 IF b(xn,nyo)=160 THEN SOUND 1,1500,
10,5:zz=3:p=p+1:b(xn,nyo)=163:sc=sc+30:P
EN #1,14:LOCATE #1,2,25:PRINT #1,"Punkte
: ";sc;
1590 RETURN
1595 REM ---gewitter---
1600 ENV 6,4,0,10,15,-1,75
1605 ENT 6,1,0,10,10,10,1,10,-10,1,10,10
,1,10,-10,1
1610 SOUND 7,300,350,15,6,6,15
1615 INK 0,25:FOR i=1 TO 23:NEXT i:INK 0
,0
1620 FOR i=1 TO 50:NEXT i
1625 INK 0,24:FOR i=1 TO 50:NEXT i:INK 0
,0:RETURN

1630 REM ---ende---
1635 MODE 1:INK 0,1:INK 1,24:PEN 1:PAPER
 0:BORDER 1:END
```

```
CHECK V1:PRUEFSUMMEN:

ZEILENNR.:  SUMMEN
    5-   10: 28353
   15-   20: 14278
   25-   30: 17061
   35-   40: 32706
   45-   50: 25199
   55-   60: 47309
   65-   70: 54355
   75-   80: 25619
   85-   90: 62588
   95-  100: 82665
  105-  110: 70415
  115-  120: 51807
  125-  130: 26782
  135-  140: 27659
  145-  150: 98793
  155-  160: 48323
  165-  170: 35123
  175-  180: 42092
  185-  190: 34433
  195-  200:102083
  205-  210:112678
  215-  220: 51141
  225-  230: 54809
  235-  240: 47217
  245-  250: 63081
  255-  260: 52102
  265-  270: 60842
  275-  280: 71924
  285-  290:  4496
  295-  300: 20778
  305-  310: 66506
  315-  320: 11063
  325-  330: 58238
  335-  340: 72071
  345-  350: 53323
  355-  360: 20675
  365-  370: 94192
  375-  380: 60523
  385-  390: 63330
  395-  400: 18907
  405-  410: 46836
  415-  420: 57706
  425-  430: 62206
  435-  440: 50854
  445-  450: 65449
  455-  460: 52149
  465-  470: 56504
  475-  480: 33011
  485-  490: 23074
  495-  500: 27297
  505-  510: 56527
  515-  520: 58278
  525-  530: 69737
  535-  540: 52930
  545-  550: 47653
  555-  560:123337
  565-  570: 12484
  575-  580: 74597
  585-  590: 47737
  595-  600: 62124
  605-  610: 36814
  615-  620: 59266
  625-  630:  3496
  635-  640: 28637
  645-  650: 78069
  655-  660: 49330
  665-  670: 82117
  675-  680: 60731
  685-  690: 68025
  695-  700: 36056
  705-  710: 21648
  715-  720: 48860
  725-  730: 52090
  735-  740: 44498
  745-  750:109742
  755-  760: 60461
  765-  770: 67605
  775-  780: 36056
  785-  790: 66920
  795-  800: 78069
  805-  810: 22859
  815-  820: 13320
  825-  830: 61001
  835-  840: 57158
  845-  850: 12770
  855-  860: 22046
  865-  870: 97169
  875-  880: 36125
  885-  890: 34129
  895-  900:  5821
  905-  910: 57298
  915-  920: 69120
  925-  930: 89275
  935-  940: 13307
  945-  950: 47689
  955-  960: 63539
  965-  970: 57470
  975-  980: 12875
  985-  990: 41234
  995- 1000: 27203
 1005- 1010: 85993
 1015- 1020: 97169
 1025- 1030: 36125
 1035- 1040: 34129
 1045- 1050:  5821
 1055- 1060: 57142
 1065- 1070: 64080
 1075- 1080: 82976
 1085- 1090: 13307
 1095- 1100: 47689
 1105- 1110: 63434
 1115- 1120: 57314
 1125- 1130: 12875
 1135- 1140: 37045
 1145- 1150: 23964
 1155- 1160: 12109
 1165- 1170: 12303
 1175- 1180: 18735
 1185- 1190: 13553
 1195- 1200: 48423
 1205- 1210: 66835
 1215- 1220: 60542
 1225- 1230: 76489
 1235- 1240: 95380
 1245- 1250: 38893
 1255- 1260: 13782
 1265- 1270: 21836
 1275- 1280: 35149
 1285- 1290: 21339
 1295- 1300: 46908
 1305- 1310: 25895
 1315- 1320: 63664
 1325- 1330: 33286
 1335- 1340: 42204
 1345- 1350: 16983
 1355- 1360: 12828
 1365- 1370: 22300
 1375- 1380: 82349
 1385- 1390: 33057
 1395- 1400:  8022
 1405- 1410: 32157
 1415- 1420: 34291
 1425- 1430: 36682
 1435- 1440:  9300
 1445- 1450: 81217
 1455- 1460: 33172
 1465- 1470: 24455
 1475- 1480: 45363
 1485- 1490: 46714
 1495- 1500: 57700
 1505- 1510: 31076
 1515- 1520: 21080
 1525- 1530: 16522
 1535- 1540: 63164
 1545- 1550: 37071
 1555- 1560: 24768
 1565- 1570: 21709
 1575- 1580: 24551
 1585- 1590: 37962
 1595- 1600: 20766
 1605- 1610: 47491
 1615- 1620: 45362
 1625- 1630: 44039
 1635-      : 21525
------------------------------
GESAMTCHECKSUMME:   3362
```

MIDNIGHT *THE END*

HORROR CAVES



Dieses Spiel bringt Ihnen den absoluten HORROR!!!

Die metallverarbeitende Industrie ist in Schwierigkeiten geraten: ihre Rohstoffvorräte gehen zur Neige, und neue Quellen müssen erschlossen werden. Doch die umgehend angestellten Bodenanalysen ergeben, daß in den Pyrenäen ein reicher Vorrat an Eisenerz lagert, der nur darauf wartet, abgetragen zu werden. Sofort wird ein erfahrener Mann mit dieser Aufgabe betraut, doch ihre Durchführung gestaltet sich schwieriger, als man erwartet hatte . . .

Denn in den Höhlen der Pyrenäen lauern blutrünstige Geister auf den nichtsahnenden Bergbauspezialisten, und mit dieser Ausgeburt des Bösen ist wahrlich nicht zu spaßen. Doch nicht nur diese Geister machen das Einsammeln der so wichtigen Erze zu einer lebensgefährlichen Aufgabe; erschwerend kommt noch hinzu, daß bei der Arbeit „Untertage" nur ein knapper Vorrat an Sauerstoff zur Verfügung steht und daß man sich nur allzu leicht in den unterirdischen Höhlen verirren kann! (bez.)

Einige wichtige Tips:
Sie können den Schwierigkeitsgrad des Spiels bestimmen (Profi, Amateur, Anfänger); dies wirkt sich sowohl auf die Geschwindigkeit der Geister als auch auf die Höhe Ihres Luftvorrates aus. Haben Sie das Minimum der zum Überleben wichtigen Luftreserven erreicht, so werden Sie durch einen Piepston gewarnt. Sie können neue Luft tanken, wenn Sie bis dahin schon 300 Punkte gesammelt und die Controll-Taste (CTRL) und „Q" gedrückt haben. Wenn der Geist Ihnen zu nahe „auf die Pelle rückt", können Sie ihm durch schlagartige Kursänderung ausweichen.

m,o	– Variablen für Spielbewegung
x,y,g	– Variablen für Geistersteuerung und Geschwindigkeit
ba	– Variablen für Bildschirmaufbau und Graphik
a,b,c	– Variablen für Graphik
k,t,l,m	– Variable zur Bestimmung von Bildschirmteilen
a$,n$,w	– Tastaturabfrage
Lu,Pu	– Variablen für Luft und Punkte
ae,ab	– Zufallsvariablen für Position des Erzes.

Grafik und Musik werden mit Hilfe von „Read" und „Data" erstellt.

Und noch eins:
Wenn Sie einen der Gänge verlassen wollen, müssen Sie den Spieler bis zum Ende des Ganges bewegen und kommen dann automatisch in den nächsten Gang. Gesteuert wird das Spiel mittels Cursor-Tasten.

PS: Wenn ihnen das Spiel zu leicht werden sollte, müssen Sie nur in den Zeilen 890 und 900 „x=m and" herausnehmen.

HORROR CAVES — DAS LISTING

```
10 REM ******************************
20 REM ***      HORROR-CAVES       ***
30 REM ***                         ***
40 REM ***  WRITTEN AND DESIGNED   ***
50 REM ***                         ***
60 REM ***          BY...          ***
70 REM ***                         ***
80 REM ***    UDO BREHMER     1985 ***
90 REM ******************************
100 REM
110 REM ++++++++++++++++++++++++++
120 REM ++++  Spieleinleitung  ++++
130 REM ++++++++++++++++++++++++++
140 REM
150 INK 0,0:INK 1,24:INK 2,2:INK 3,6:BOR
DER 1:MODE 1
160 CLS:PEN 1:PRINT "Sind Sie :"
170 PRINT:PRINT "Anfaenger......./ 1 /"
180 PRINT:PRINT "Amateur........./ 2 /"
190 PRINT:PRINT "Profi.........../ 3 /"
200 PRINT:PEN 2:INPUT "Ihre Wahl";w
210 IF w<1 OR w>3 THEN 200
220 INK 3,6:GOSUB 2720
230 SYMBOL AFTER 32
240 SYMBOL 207,239,231,219,53,204,138,23
5,236
250 SYMBOL 224,24,60,102,126,199,187,255
,126
260 SYMBOL 143,96,156,68,72,38,18,44,112
270 SYMBOL 251,184,28,72,124,20,24,104,2
4
280 SYMBOL 250,26,56,18,62,40,24,22,24
290 SYMBOL 248,186,124,40,56,16,40,40,10
8
300 SYMBOL 249,56,56,108,186,16,40,40,10
8
310 DIM LA(40,25):M=25:O=22:lu=200
320 SPEED KEY 1,1
330 ON BREAK GQSUB 340:GOTO 350
340 SPEED KEY 20,2:CLS:PEN 1:END
350 GOSUB 2210
360 GOTO 1020
370 REM
```

```
380 REM +++++++++++++++++++++++++++++
390 REM +++++   Hauptroutinen   +++++
400 REM +++++++++++++++++++++++++++++
410 REM
420 INK 1,24:INK 2,2:INK 3,6:INK 0,0:LOC
ATE m,o:PRINT CHR$(249)
430 GOSUB 2600
440 LOCATE 1,24:PRINT"Punkte:";pu
450 IF LA(M,O)=0 THEN 680
460 IF LA(M,O)=1 THEN M=2:GOTO 1120
470 IF LA(M,O)=2 THEN O=2:GOTO 1300
480 IF LA(M,O)=3 THEN O=23:GOTO 1020
490 IF LA(M,O)=4 THEN M=38:GOTO 1220
500 IF LA(M,O)=5 THEN M=3:GOTO 1300
510 IF LA(M,O)=6 THEN O=2:GOTO 1680
520 IF LA(M,O)=7 THEN O=23:GOTO 1300
530 IF LA(M,O)=8 THEN M=38:GOTO 1600
540 IF LA(M,O)=9 THEN M=3:GOTO 1680
550 IF LA(M,O)=10 THEN M=3:GOTO 1780
560 IF LA(M,O)=11 THEN M=38:GOTO 1680
570 IF LA(M,O)=12 THEN O=2:GOTO 2060
580 IF LA(M,O)=13 THEN M=2:GOTO 1880
590 IF LA(M,O)=14 THEN M=38:GOTO 1780
600 IF LA(M,O)=15 THEN O=23:GOTO 1500
610 IF LA(M,O)=16 THEN O=2:GOTO 1880
620 IF LA(M,O)=17 THEN M=38:GOTO 1400
630 IF LA(M,O)=18 THEN M=2:GOTO 1500
640 IF LA(M,O)=19 THEN O=23:GOTO 1120
650 IF LA(M,O)=20 THEN O=2:GOTO 1400
660 IF LA(M,O)=21 THEN M=38:GOTO 1020
670 IF LA(M,O)=22 THEN O=23:GOTO 1780
680 IF LA(M,O)=23 THEN M=38:GOTO 1980
690 IF LA(M,O)=24 THEN M=3:GOTO 2060
700 REM
710 REM ++++++++++++++++++++++++
720 REM ++++ Spielerbewegung ++++
730 REM ++++++++++++++++++++++++
740 REM
750 A$=INKEY$
760 IF A$=CHR$(242) AND LA(M-1,O)<>50 TH
EN M=M-1:LOCATE M,O:PRINT CHR$(251):LOCA
TE M+1,O:PRINT" "
770 IF A$=CHR$(243) AND LA(M+1,O)<>50 TH
EN M=M+1:LOCATE M,O:PRINT CHR$(250):LOCA
TE M-1,O:PRINT" "
780 IF A$=CHR$(240) AND LA(M,O-1)<>50 TH
EN O=O-1:LOCATE M,O:PRINT CHR$(248):LOCA
TE M,O+1:PRINT" "
790 IF A$=CHR$(241) AND LA(M,O+1)<>50 TH
EN O=O+1:LOCATE M,O:PRINT CHR$(249):LOCA
TE M,O-1:PRINT" "
800 IF la(m,o)=40 THEN la(m,o)=0:GOSUB 2
660
810 REM
820 REM ++++++++++++++++++++++++++
830 REM +++++      Geister     +++++
840 REM ++++++++++++++++++++++++++
850 REM
860 g=g+1:IF g=4-w THEN g=0:GOTO 870 ELS
E GOTO 1010
870 IF x>m AND la(x-1,y)=0 THEN x=x-1:LO
CATE x,y:PRINT CHR$(224):LOCATE x+1,y:PR
INT " "
880 IF x<m AND la(x+1,y)=0 THEN x=x+1:LO
CATE x,y:PRINT CHR$(224):LOCATE x-1,y:PR
INT " "
890 IF y<o AND m=x AND la(x,y+1)=0 THEN
y=y+1:LOCATE x,y:PRINT CHR$(224):LOCATE
x,y-1:PRINT " "
900 IF y>o AND m=x AND la(x,y-1)=0 THEN
y=y-1:LOCATE x,y:PRINT CHR$(224):LOCATE
x,y+1:PRINT " "
910 IF m=x AND o=y THEN SPEED KEY 20,2:S
OUND 1,500,50,4,2,2,28:FOR t=1 TO 3000:N
EXT:RUN
920 REM
930 REM +++++++++++++++++++++++++
940 REM +++++   Luftabnahme   +++++
950 REM +++++++++++++++++++++++++
960 REM
970 lu=lu-w:LOCATE 1,23:PRINT"Luft:";lu
980 IF lu<60 THEN SOUND 1,300,4,4:BORDER
 3
990 IF INKEY(67)=128 AND pu>=300 THEN lu
=200:pu=pu-300:BORDER 0:GOSUB 2690
1000 IF lu<1 THEN FOR t=1 TO 3000:NEXT:R
UN
1010 GOTO 450
1020 REM ******************************1
1030 ERASE LA:DIM LA(40,25)
1040 CLS:PEN 2
1050 x=21:y=11
1060 FOR BA=25 TO 10 STEP-1:LOCATE 20,BA
:PRINT CHR$(207):LA(20,BA)=50:NEXT
1070 FOR BA=25 TO 15 STEP-1:LOCATE 30,BA
:PRINT CHR$(207):LA(30,BA)=50:NEXT
1080 FOR BA=20 TO 40:LOCATE BA,10:PRINT
CHR$(207):LA(BA,10)=50:NEXT
1090 FOR BA=30 TO 40:LOCATE BA,15:PRINT
CHR$(207):LA(BA,15)=50:NEXT
1100 FOR BA=11 TO 14:LA(40,BA)=1:NEXT:FO
R BA=20 TO 30:LA(BA,25)=2:NEXT
1110 PEN 1:GOTO 420
1120 REM ******************************2
1130 ERASE LA:DIM LA(40,25)
1140 CLS:PEN 2
1150 x=29:y=11
1160 FOR BA=1 TO 30:LOCATE BA,10:PRINT C
HR$(207):LA(BA,10)=50:NEXT
1170 FOR BA=1 TO 20:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
1180 FOR BA=10 TO 25:LOCATE 30,BA:PRINT
CHR$(207):LA(30,BA)=50:NEXT
1190 FOR BA=15 TO 25:LOCATE 20,BA:PRINT
CHR$(207):LA(20,BA)=50:NEXT
1200 FOR BA=11 TO 14:LA(1,BA)=21:NEXT:FO
R BA=21 TO 29:LA(BA,25)=20:NEXT
1210 PEN 1:GOTO 420
1220 REM ******************************3
1230 ERASE LA:DIM LA(40,25)
1240 CLS:PEN 3
1250 x=5:y=14
1260 FOR BA=4 TO 40:LOCATE BA,10:PRINT C
HR$(207):LOCATE BA,15:PRINT CHR$(207):LA
(BA,10)=50:LA(BA,15)=50:NEXT
1270 FOR BA=10 TO 15:LOCATE 3,BA:PRINT C
HR$(207):LA(3,BA)=50:NEXT
1280 FOR BA=11 TO 14:LA(40,BA)=5:NEXT
1290 PEN 1:GOTO 420
1300 REM ******************************4
1310 ERASE LA:DIM LA(40,25)
```



```
1320 CLS:PEN 3
1330 x=25:y=20
1340 FOR BA=1 TO 25:LOCATE 30,BA:PRINT C
HR$(207):LA(30,BA)=50:NEXT
1350 FOR BA=1 TO 10:LOCATE 20,BA:PRINT C
HR$(207):LA(20,BA)=50:NEXT
1360 FOR BA=1 TO 20:LOCATE BA,10:PRINT C
HR$(207):LOCATE BA,15:PRINT CHR$(207):LA
(BA,10)=50:LA(BA,15)=50:NEXT
1370 FOR BA=15 TO 25:LOCATE 20,BA:PRINT
CHR$(207):LA(20,BA)=50:NEXT
1380 FOR BA=11 TO 14:LA(1,BA)=4:NEXT:FOR
 BA=21 TO 29:LA(BA,1)=3:LA(BA,25)=6:NEXT
1390 PEN 1:GOTO 420
1400 REM ******************************5
1410 ERASE LA:DIM LA(40,25)
1420 CLS:PEN 3
1430 x=21:y=19
1440 FOR BA=1 TO 20:LOCATE 20,BA:PRINT C
HR$(207):LA(20,BA)=50:NEXT
1450 FOR BA=1 TO 15:LOCATE 30,BA:PRINT C
HR$(207):LA(30,BA)=50:NEXT
1460 FOR BA=20 TO 40:LOCATE BA,20:PRINT
CHR$(207):LA(BA,20)=50:NEXT
1470 FOR BA=30 TO 40:LOCATE BA,15:PRINT
CHR$(207):LA(BA,15)=50:NEXT
1480 FOR BA=21 TO 29:LA(BA,1)=19:NEXT:FO
R BA=16 TO 19:LA(39,BA)=18:NEXT
1490 PEN 1:GOTO 420
1500 REM ******************************6
1510 ERASE LA:DIM LA(40,25)
1520 CLS:PEN 3
1530 x=29:y=16
1540 FOR BA=1 TO 30:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
1550 FOR BA=1 TO 20:LOCATE BA,20:PRINT C
HR$(207):LA(BA,20)=50:NEXT
1560 FOR BA=20 TO 25:LOCATE 20,BA:PRINT
CHR$(207):LA(20,BA)=50:NEXT
1570 FOR BA=15 TO 25:LOCATE 30,BA:PRINT
CHR$(207):LA(30,BA)=50:NEXT
1580 FOR BA=16 TO 19:LA(1,BA)=17:NEXT:FO
R BA=21 TO 29:LA(BA,25)=16:NEXT
1590 PEN 1:GOTO 420
1600 REM ******************************7
1610 ERASE LA:DIM LA(40,25)
1620 CLS:PEN 2
1630 x=11:y=16
1640 FOR BA=10 TO 40:LOCATE BA,15:PRINT
CHR$(207):LOCATE BA,20:PRINT CHR$(207):L
A(BA,15)=50:LA(BA,20)=50:NEXT
1650 FOR BA=16 TO 19:LOCATE  10,BA:PRINT
 CHR$(207):LA(10,BA)=50:NEXT
1660 FOR BA=16 TO 19:LA(40,BA)=9:NEXT
1670 PEN 1:GOTO 420
1680 REM ******************************8
1690 ERASE LA:DIM LA(40,25)
1700 CLS:PEN 2
1710 x=20:y=19
1720 FOR BA=1 TO 40:LOCATE BA,20:PRINT C
HR$(207):LA(BA,20)=50:NEXT
1730 FOR BA=1 TO 20:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
1740 FOR BA=15 TO 1 STEP-1:LOCATE 20,BA:
PRINT CHR$(207):LOCATE 30,BA:PRINT CHR$(
207):LA(30,BA)=50:LA(20,BA)=50:NEXT
1750 FOR BA=30 TO 40:LOCATE BA,15:PRINT
CHR$(207):LA(BA,15)=50:NEXT
1760 FOR BA=21 TO 29:LA(BA,1)=7:NEXT:FOR
 BA=16 TO 19:LA(1,BA)=8:LA(40,BA)=10:NEX
T
1770 PEN 1:GOTO 420
1780 REM ******************************9
1790 ERASE LA:DIM LA(40,25)
1800 CLS:PEN 2
1810 x=20:y=19
1820 FOR BA=1 TO 40:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
1830 FOR BA=1 TO 20:LOCATE BA,20:PRINT C
HR$(207):LA(BA,20)=50:NEXT
1840 FOR BA=20 TO 25:LOCATE 20,BA:PRINT
CHR$(207):LA(20,BA)=50:LOCATE 30,BA:PRIN
T CHR$(207):LA(30,BA)=50:NEXT
1850 FOR BA=30 TO 40:LOCATE BA,20:PRINT
CHR$(207):LA(BA,20)=50:NEXT
1860 FOR BA=16 TO 19:LA(1,BA)=11:LA(40,B
A)=13:NEXT:FOR BA=20 TO 30:LA(BA,25)=12:
NEXT
1870 PEN 1:GOTO 420
1880 REM *****************************10
1890 ERASE LA:DIM LA(40,25)
1900 CLS:PEN 2
1910 x=29:y=19
1920 FOR BA=1 TO 30:LOCATE BA,20:PRINT C
HR$(207):LA(BA,20)=50:NEXT
1930 FOR BA=1 TO 20:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
1940 FOR BA=1 TO 15:LOCATE 20,BA:PRINT C
HR$(207):LA(20,BA)=50:NEXT
1950 FOR BA=20 TO 1 STEP-1:LOCATE 30,BA:
PRINT CHR$(207):LA(30,BA)=50:NEXT
1960 FOR BA=16 TO 19:LA(1,ba)=14:NEXT:FO
R BA=21 TO 29:LA(BA,1)=15:NEXT
1970 PEN 1:GOTO 420
1980 REM ****************************11
1990 ERASE LA:DIM LA(40,25)
2000 CLS:PEN 3
2010 x=11:y=19
2020 FOR BA=10 TO 40:LOCATE BA,15:PRINT
CHR$(207):LOCATE BA,20:PRINT CHR$(207):L
A(BA,15)=50:LA(BA,20)=50:NEXT
2030 FOR BA=16 TO 19:LOCATE  10,BA:PRINT
 CHR$(207):LA(10,BA)=50:NEXT
2040 FOR BA=16 TO 19:LA(40,BA)=24:NEXT
2050 PEN 1:GOTO 420
2060 REM ****************************12
2070 ERASE LA:DIM LA(40,25)
2080 CLS:PEN 3
2090 x=29:y=19
2100 FOR BA=1 TO 30:LOCATE BA,20:PRINT C
HR$(207):LA(BA,20)=50:NEXT
2110 FOR BA=1 TO 20:LOCATE BA,15:PRINT C
HR$(207):LA(BA,15)=50:NEXT
2120 FOR BA=15 TO 1 STEP-1:LOCATE 20,BA:
PRINT CHR$(207):LA(20,BA)=50:NEXT
2130 FOR BA=20 TO 1 STEP-1:LOCATE 30,BA:
PRINT CHR$(207):LA(30,BA)=50:NEXT
2140 FOR BA=21 TO 29:LA(BA,1)=22:NEXT:FO
R BA=16 TO 19:LA(1,BA)=23:NEXT
2150 PEN 1:GOTO 420
```

```
2160 REM
2170 REM ++++++++++++++++++++++++++
2180 REM +++++      Graphik     +++++
2190 REM ++++++++++++++++++++++++++
2200 REM
2210 CLS
2220 PLOT 20,300,0
2230 FOR ba=1 TO 54
2240 READ a,b,c
2250 DRAWR a,b,c
2260 SOUND 1,50,1,7:SOUND 1,0,2,0
2270 NEXT
2280 DATA 0,-100,3,0,150,3,0,-80,3,50,0,
3,0,-150,3,0,200,3
2290 DATA 28,-50,0,-25,-40,3,25,-40,3,25
,40,3,-25,40,3
2300 DATA 28,0,0,0,-80,3,0,60,3,50,20,3,
3,0,0,0,-80,3,0,60,3,50,20,3,3,0,0
2310 DATA 3,0,0,-25,-40,3,25,-40,3,25,40
,3,-25,40,3,28,0,0,0,-80,3,0,60,3,50,20,
3,3,-40,0,20,2,3,20,-3,3,10,1,3
2320 DATA 60,100,0,-50,-100,3,80,-120,3,
-20,160,0,-40,-40,3,40,-40,3,0,80,3,0,-8
0,3,10,0,3
2330 DATA 3,80,0,25,-80,3,25,80,3,5,-20,
0,40,0,3,-20,20,3,-20,-20,3,30,-60,3
2340 DATA 53,80,0,-40,-26,3,40,-26,3,-40
,-26,3
2350 SPEED INK 20,20
2360 INK 3,0,26:INK 0,0:BORDER 0
2370 LOCATE 9,24:PRINT"Press any key to
play"
2380 REM
2390 REM ++++++++++++++++++++++++++
2400 REM +++++      Musik       +++++
2410 REM ++++++++++++++++++++++++++
2420 REM
2430 RESTORE 2500
2440 FOR mu=1 TO 98
2450 READ k,t,l
2460 SOUND k,t,1.1*l,7:SOUND 1,0,2,0
2470 a$=INKEY$:IF a$<>"" THEN RETURN
2480 NEXT
2490 RESTORE 2500:GOTO 2440
2500 DATA 1,239,25,1,239,25,1,239,25,1,2
39,12.5,1,239,12.5,1,239,12.5,1,239,12.5
,1,239,12.5,1,239,12.5,1,239,25
2510 DATA 1,319,25,1,284,25,1,268,25,1,2
53,12.5,2,358,12.5,1,253,12.5,2,358,12.5
,1,284,12.5,2,358,12.5,1,319,12.5,2,358,
12.5
2520 DATA 1,253,12.5,2,358,12.5,1,253,12
.5,2,358,12.5,1,284,12.5,2,358,12.5,1,31
9,12.5,2,358,12.5
2530 DATA 1,239,12.5,2,379,12.5,1,239,12
.5,2,379,12.5,1,284,12.5,2,379,12.5,1,31
9,12.5,2,379,12.5,1,239,12.5,2,379,12.5,
1,239,12.5,2,379,12.5,1,284,12.5,2,379,1
2.5,1,319,12.5,2,379,12.5
2540 DATA 1,253,25,1,284,12.5,1,319,25,1
,284,12.5,1,319,12.5,1,319,12.5,1,284,25
,1,319,12.5,1,379,25,1,0,12,1,319,12.5
2550 DATA 1,253,12.5,2,358,12.5,1,253,12
.5,2,358,12.5,1,284,12.5,2,358,12.5,1,31
9,12.5
2560 DATA 2,358,12.5,1,253,12.5,2,358,12
.5,1,253,12.5,2,358,12.5,1,284,12.5,2,35
8,12.5,1,319,12.5,2,358,12.5
2570 DATA 1,239,12.5,2,379,12.5,1,239,12
.5,2,379,12.5,1,284,12.5,2,379,12.5,1,31
9,12.5,2,379,12.5,1,239,12.5
2580 DATA 2,379,12.5,1,239,12.5,2,379,12
.5,1,284,12.5,2,379,12.5,1,319,12.5,2,37
9,12.5
2590 DATA 1,253,25,1,284,12.5,1,319,25,1
,284,12.5,1,319,12.5,1,358,12.5,1,379,25
,1,426,12.5,1,478,25,1,0,80
2600 REM ***********************
2610 FOR e=1 TO 10
2620 ae=INT(RND(1)*39)+1
2630 ab=INT(RND(1)*24)+1
2640 IF la(ae,ab)<>50 THEN LOCATE ae,ab:
PRINT CHR$(143):la(ae,ab)=40 ELSE 2620
2650 NEXT:RETURN
2660 REM ****************
2670 SOUND 1,200,10,4
2680 pu=pu+100
2690 LOCATE 1,24:PRINT "Punkte:";pu
2700 RETURN
2710 REM
2720 REM *****************************
2730 REM ***     Spielanweisung      ***
2740 REM *****************************
2750 REM
2760 SPEED KEY 20,2:CLS
2770 LOCATE 1,2:INPUT "Name";n$
2780 PRINT "Einen Moment.......":FOR t=1
 TO 2000:NEXT
2790 PEN 3:PRINT n$:PEN 1
2800 PRINT "In unserem Archiv haben wir
gesehen,dass Sie der faehigste Mann sind
,um diesen  Auftrag auszufuehren !!"
2810 PEN 2:PRINT:PRINT"Ihre Aufgabe:":PE
N 3
2820 PRINT"-------------":PEN 1
2830 PRINT:PRINT:PRINT" Uns gehen langsa
m, aber sicher die Vor-raete an Metallen
 aus!  Deshalb sammeln Sie in den Hoehle
n der Pyrenaen Erze,   um uns die Metall
industrie aufrecht zu  erhalten."
2840 PEN 3:PRINT:PRINT"Verlieren Sie nic
ht die Orientierung !!!":PEN 1
2850 PRINT:PRINT"Viel Glueck !!!"
2860 LOCATE 9,24:PRINT "Press any key to
 go on"
2870 a$=INKEY$:IF a$<>"" THEN RETURN ELS
E 2870
```

CHECK V1:PRUEFSUMMEN:

ZEILENNR.:	SUMMEN	ZEILENNR.	SUMMEN	ZEILENNR.	SUMMEN
		350- 360:	2210	750- 760:	37418
		370- 380:	21593	770- 780:	68330
		390- 400:	52871	790- 800:	48522
10- 20:	41506	410- 420:	1773	810- 820:	18970
30- 40:	32487	430- 440:	21810	830- 840:	38281
50- 60:	14405	450- 460:	67191	850- 860:	24938
70- 80:	26282	470- 480:	88911	870- 880:	61020
90- 100:	22444	490- 500:	87438	890- 900:	108252
110- 120:	50076	510- 520:	85274	910- 920:	51045
130- 140:	18970	530- 540:	87301	930- 940:	43956
150- 160:	34404	550- 560:	96288	950- 960:	18970
170- 180:	39282	570- 580:	84637	970- 980:	75372
190- 200:	39002	590- 600:	100605	990- 1000:	46907
210- 220:	27050	610- 620:	90241	1010- 1020:	23502
230- 240:	37655	630- 640:	93507	1030- 1040:	12763
250- 260:	51074	650- 660:	96135	1050- 1060:	64364
270- 280:	33226	670- 680:	99863	1070- 1080:	91226
290- 300:	39770	690- 700:	41447	1090- 1100:	63490
310- 320:	60376	710- 720:	45517	1110- 1120:	25070
330- 340:	11915	730- 740:	16519	1130- 1140:	12763

```
1150- 1160: 41366
1170- 1180: 69510
1190- 1200: 65728
1210- 1220: 25102
1230- 1240: 12768
1250- 1260: 18697
1270- 1280: 72211
1290- 1300: 25134
1310- 1320: 12768
1330- 1340: 43591
1350- 1360: 43203
1370- 1380: 72877
1390- 1400: 25166
1410- 1420: 12768
1430- 1440: 42858
1450- 1460: 70115
1470- 1480: 65220
1490- 1500: 25198
1510- 1520: 12768
1530- 1540: 41811
1550- 1560: 69340
1570- 1580: 66081
1590- 1600: 25230
1610- 1620: 12763
1630- 1640: 25813
1650- 1660: 79159
1670- 1680: 25262
1690- 1700: 12763
1710- 1720: 42303
1730- 1740: 74074
1750- 1760: 75061
1770- 1780: 25294
1790- 1800: 12763
1810- 1820: 41933
1830- 1840: 56491
1850- 1860: 89882
1870- 1880: 25223
1890- 1900: 12763
1910- 1920: 42226
1930- 1940: 65744
1950- 1960: 80179
1970- 1980: 23897
1990- 2000: 12768
2010- 2020: 25858
2030- 2040: 80218
2050- 2060: 23928
2070- 2080: 12768
2090- 2100: 42226
2110- 2120: 84050
2130- 2140: 78631
2150- 2160:  2764
2170- 2180: 37307
2190- 2200: 18970
2210- 2220:  1709
2230- 2240: 16895
2250- 2260: 16354
2270- 2280: 60637
2290- 2300: 91081
2310- 2320: 87953
2330- 2340: 81298
2350- 2360:  7642
2370- 2380: 40719
2390- 2400: 35739
2410- 2420: 20260
2430- 2440: 10024
2450- 2460: 50201
2470- 2480: 22351
2490- 2500:  4118
2510- 2520: 69804
2530- 2540: 88653
2550- 2560: 70449
2570- 2580: 71898
2590- 2600: 59459
2610- 2620: 26360
2630- 2640: 84726
2650- 2660:  8034
2670- 2680: 14665
2690- 2700: 22738
2710- 2720: 21100
2730- 2740: 50128
2750- 2760:  2964
2770- 2780: 73990
2790- 2800: 42896
2810- 2820: 35318
2830- 2840: 52822
2850- 2860: 56207
2870-      : 28368
---------------------------
GESAMTCHECKSUMME:   41005
```

Für die Freunde der Selbstverteidigung

„Thai-Boxing"

TI-99-Fans aufgepaßt! Dieses tolle Spiel stellt hohe Anforderungen an Körperbeherrschung, Konzentration und Reaktionsvermögen!

Nach dem Programmstart werdet ihr aufgefordert, einen Schwierigkeitsgrad in Form von drei wählbaren Gegnern sowie den eigenen Namen einzugeben. Danach findet ihr euch als weißen Thai-Boxer gemeinsam mit dem schwarzen Gegner in der Kampfarena wieder. Ihr müßt nun versuchen, dem Gegener möglichst viele Schläge und Tritte beizubringen. Doch der Kontrahent wehrt sich, denn er beherrscht alle Tritte und Schläge genauso perfekt wie ihr selbst.

Achtung: Bei Schwierigkeitsgrad 1 benötigt ihr oder der Gegener 5 Schläge oder Tritte, um einen von beiden auf die Matte zu legen. Bei den weiteren Schwierigkeitsgraden benötigt der Gegner nur 3 oder 4 Treffer, um euch zu bezwingen. Jedesmal, wenn ihr den Gegner auf den Boden befördert, wird ein Punkt gutgeschrieben. Sieger ist derjenige, der zuerst 10 Punkte erreicht.

Thai-Boxing wird über die Tastatur gesteuert. Die Tastenfunktionen können aus der Spielanleitung des Programmes entnommen werden. Zum Spiel muß die Alpha Lock-Taste gedrückt sein.

Und nun ran an die Tasten!

```
10 CALL CLEAR :: CALL SPRITE(#1,84,16,30
,50,#2,72,2,30,80,#3,65,16,30,110,#4,73,
2,30,140)
20 CALL SPRITE(#5,66,16,50,50,#6,79,2,70
,70,#7,88,16,90,90,#8,73,2,110,110)
30 CALL SPRITE(#9,78,16,130,130,#10,71,2
,150,150)
40 DISPLAY AT(1,1):"INTERFACE-SOFTWARE P
RESENTS:" :: DISPLAY AT(24,3):"CO.1985 B
Y: DIRK BLUDAU"
50 DISPLAY AT(8,14)SIZE(15):"PRESS ANY K
EY !"
60 CALL MAGNIFY(1):: FOR I=1 TO 10 :: GO
SUB 70 :: NEXT I :: CALL MAGNIFY(2):: FO
R I=1 TO 10 :: GOSUB 70 :: NEXT I :: GOT
O 60
70 CALL KEY(0,K,S):: IF S<>0 THEN CALL C
LEAR :: CALL DELSPRITE(ALL):: GOTO 80 EL
SE RETURN
80 DISPLAY AT(3,3):"SPIELANLEITUNG ? <J>
<N>" :: FOR J=1 TO 100 :: NEXT J
90 CALL KEY(0,K,S):: IF S=0 THEN 90 ELSE
 IF K=74 THEN 1200 ELSE 100
100 Q,R=0 :: CALL CLEAR :: DISPLAY AT(3,
3):"SCHWIERIGKEITSGRAD ?" :: DISPLAY AT(
10,3):"1 = LEICHT (HO-CHI-MINH)"
```

Texas Instruments

Dieses Spiel ist lauffähig auf dem TI-99/4a Extended Basic



```
110 DISPLAY AT(12,3):"2 = NORMAL (CHANG-
PUANG)" :: DISPLAY AT(14,3):"3 = SCHWER
(CHAM-PI-IN)" :: ACCEPT AT(20,3)VALIDATE
("123")BEEP:C
120 IF C=1 THEN CW=5 ELSE IF C=2 THEN CW
=4 ELSE CW=3
130 CALL CLEAR :: CC=C*10 :: IF C=1 THEN
 CH=10 ELSE IF C=2 THEN CH=28 ELSE CH=46
140 DISPLAY AT(3,3):"GIB DEINEN NAMEN EI
N !" :: ACCEPT AT(10,3)VALIDATE(UALPHA)B
EEP SIZE(12):X$
150 IF C=1 THEN Y$="HO-CHI-MINH" ELSE IF
 C=2 THEN Y$="CHANG-PUANG" ELSE Y$="CHAM
-PI-IN"
160 CALL CLEAR :: DISPLAY AT(4,8)SIZE(12
):X$ :: DISPLAY AT(5,8)SIZE(12):Y$
170 CALL CHAR(33,"FFFFFFFFFFFFFFFF",34,"
00000000000000FF",35,"8080808080808080",
36,"FF00000000000000",37,"01010101010101
01")
180 CALL CHAR(40,"003C5A7E663C1818")
190 DATA 001C1A1C183C3F3C3F3C3C386CC6864
3000000000000008000000000000000000,001C1A
1C183C3F3C3F3C3C181818181C00000000000800
0
000000000000000000
200 DATA 00000000000100000000000000000000
00038583818 3CFC3CFC3C3C1C366361C2,000000
00000100000000000000000000000385838183CFC
3CFC3C3C1818181838
210 DATA 0000000000010305030101010306040
200E0D0E0C0E0FFE0E0E0E0C060303018,000101
010103070B07030303060C0C0600C0A0C080C8F0
C0C1FFFE0000000000
220 DATA 0101010103070B07037FFC800000000
0C0A0C080C0FCC0C1FF7E000000000000,00070B
070307FF0707070703060C0C1800000000000080C0
A0C0808080C0602040
230 DATA 0003050301130F0383FF7F000000000
00080808080C0E0D0E0E0E06030181830,030503
01033F0383FF7E00000000000080808080C0E0D0
E0C0FE3F0100000000
240 DATA 000000000101010103070B07037FFE8
000000000C0A0C888C8F0C0C0F8FC0C06,000000
00000000000000000023577F7F00000000000000
0000000000C0E1FFFF
250 DATA 0000000030513111130F03031F3F306
000000000080808080C0E0D0E0C0FE7F01,000000
0000000000000000000387FFFF00000000000000
0000000000C4EAFEFE
260 RESTORE 190 :: FOR I=88 TO 140 STEP
4 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
 :: CALL COLOR(1,7,1,2,7,1):: RANDOMIZE
270 FOR I=9 TO 16 :: Q=Q+1 :: CALL HCHAR
(I,1,33,Q):: NEXT I
280 Q=33 :: FOR I=9 TO 16 :: Q=Q-1 :: R=
R+1 :: CALL HCHAR(I,Q,33,R):: NEXT I
290 CALL HCHAR(10,4,33,26):: Q=0 :: FOR
I=8 TO 15 :: Q=Q+1 :: CALL HCHAR(I,Q,40)
:: NEXT I :: Q=33 :: FOR I=8 TO 15 :: Q=
Q-1
300 CALL HCHAR(I,Q,40):: NEXT I :: CALL
HCHAR(9,4,40,26):: CALL HCHAR(8,5,40,24)
:: CALL HCHAR(7,6,40,22)
310 CALL VCHAR(1,8,37,6):: CALL VCHAR(1,
25,35,6):: CALL HCHAR(1,9,36,16):: CALL
HCHAR(6,9,34,16)
320 DISPLAY AT(2,9)SIZE(11):"PUNKTESTAND
" :: DISPLAY AT(4,20)SIZE(2):PU :: DISPL
AY AT(5,20)SIZE(2):GU
330 GOSUB 1300
340 CALL MAGNIFY(3):: CALL SPRITE(#1,88,
16,160,30,#2,96,2,160,226):: FOR I=1 TO
100 :: NEXT I :: RI=2 :: GI=2
350 CALL SOUND(200,-3,0):: CALL SOUND(20
0,-2,0)
360 GOSUB 690 :: CALL KEY(0,K,S):: IF S=
0 THEN 360
370 IF K=83 THEN X=-4 :: Y=0 :: RI=1 ::
GOTO 440
380 IF K=68 THEN X=4 :: Y=0 :: RI=2 :: G
OTO 450
390 IF K=44 THEN 460
400 IF K=46 THEN 510
410 IF K=76 THEN 560
420 IF K=32 THEN 1030
430 GOTO 360
440 CALL PATTERN(#1,100):: CALL MOTION(#
1,-Y*7,X*7):: CALL PATTERN(#1,96):: CALL
 MOTION(#1,0,0):: GOTO 360
450 CALL PATTERN(#1,92):: CALL MOTION(#1
,-Y*7,X*7):: CALL PATTERN(#1,88):: CALL
MOTION(#1,0,0):: GOTO 360
460 CALL SOUND(100,110,0,115,0,120,0)::
IF RI=1 THEN CALL PATTERN(#1,116):: GOTO
 470 ELSE CALL PATTERN(#1,104):: GOTO 49
0
470 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 480 ELSE CALL
PATTERN(#1,96):: GOTO 360
480 CALL PATTERN(#1,96):: IF GD>0 THEN G
OTO 360 ELSE GOTO 1060
490 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 500 ELSE CALL
PATTERN(#1,88):: GOTO 360
500 CALL PATTERN(#1,88):: IF GD>0 THEN G
OTO 360 ELSE GOTO 1060
510 CALL SOUND(100,110,0,115,0,120,0)::
IF RI=1 THEN CALL PATTERN(#1,120):: GOTO
 520 ELSE CALL PATTERN(#1,108):: GOTO 54
0
520 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 530 ELSE CALL
PATTERN(#1,96):: GOTO 360
530 CALL PATTERN(#1,96):: IF GD>0 THEN G
OTO 360 ELSE GOTO 1060
540 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 550 ELSE CALL
PATTERN(#1,88):: GOTO 360
550 CALL PATTERN(#1,88):: IF GD>0 THEN G
OTO 360 ELSE GOTO 1060
560 SP=1 :: IF RI=1 THEN 610 ELSE 570
570 CALL PATTERN(#1,112):: CALL MOTION(#
1,-20,20):: FOR I=1 TO 10 :: CALL COINC(
ALL,CO):: IF CO=-1 THEN GOTO 650 ELSE 58
0
580 NEXT I
590 CALL MOTION(#1,20,20):: FOR I=1 TO 1
0 :: CALL COINC(ALL,CO):: IF CO=-1 THEN
650 ELSE 600
600 NEXT I :: CALL PATTERN(#1,88):: CALL
```



```
 MOTION(#1,0,0):: CALL POSITION(#1,E,F):
: CALL LOCATE(#1,160,F):: GOTO 360
610 CALL PATTERN(#1,124):: CALL MOTION(#
1,-20,-20):: FOR I=1 TO 10 :: CALL COINC
(ALL,CO):: IF CO=-1 THEN GOTO 670 ELSE 6
20
620 NEXT I
630 CALL MOTION(#1,20,-20):: FOR I=1 TO
10 :: CALL COINC(ALL,CO):: IF CO=-1 THEN
 670 ELSE 640
640 NEXT I :: CALL PATTERN(#1,96):: CALL
 MOTION(#1,0,0):: CALL POSITION(#1,E,F):
: CALL LOCATE(#1,160,F):: GOTO 360
650 CALL SOUND(100,110,0,115,0,120,0)::
CALL MOTION(#1,0,0):: CALL POSITION(#1,E
,F):: CALL LOCATE(#1,160,F):: CALL PATTE
RN(#1,88)
660 IF GD>0 THEN GOTO 360 ELSE 1060
670 CALL SOUND(100,110,0,115,0,120,0)::
CALL MOTION(#1,0,0):: CALL POSITION(#1,E
,F):: CALL LOCATE(#1,160,F):: CALL PATTE
RN(#1,96)
680 IF GD>0 THEN GOTO 360 ELSE 1060
690 IF GD=1 THEN 1000 ELSE IF GD=2 THEN
1020 ELSE CALL POSITION(#1,E,F):: CALL P
OSITION(#2,EE,FF):: IF F<FF THEN 700 ELS
E 710
700 IF FF-F>CC THEN 730 ELSE IF SP=1 THE
N SP=0 :: CALL PATTERN(#2,88):: GI=1 ::
GOTO 720 ELSE GOTO 720
710 IF F-FF>CC THEN 740 ELSE IF SP=1 THE
N SP=0 :: CALL PATTERN(#2,96):: GI=2 ::
GOTO 720 ELSE GOTO 720
720 O=2+C :: ZU=INT(RND*O)+1 :: IF ZU=1
THEN 750 ELSE IF ZU=2 THEN 800 ELSE IF Z
U=3 THEN 980 ELSE 850
730 GI=1 :: CALL PATTERN(#2,100):: CALL
MOTION(#2,0,-CH):: CALL PATTERN(#2,96)::
 CALL MOTION(#2,0,0):: RETURN
740 GI=2 :: CALL PATTERN(#2,92):: CALL M
OTION(#2,0,CH):: CALL PATTERN(#2,88):: C
ALL MOTION(#2,0,0):: RETURN
750 CALL SOUND(100,150,0,155,0,-2,0):: I
F GI=1 THEN CALL PATTERN(#2,116):: GOTO
760 ELSE CALL PATTERN(#2,104):: GOTO 780
760 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 770 ELSE CALL
PATTERN(#2,96):: RETURN
770 CALL PATTERN(#2,96):: IF GE=1 THEN R
ETURN ELSE GOTO 1090
780 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 790 ELSE CALL
PATTERN(#2,88):: RETURN
790 CALL PATTERN(#2,88):: IF GE=1 THEN R
ETURN ELSE GOTO 1090
800 CALL SOUND(100,150,0,155,0,-2,0):: I
F GI=1 THEN CALL PATTERN(#2,120):: GOTO
810 ELSE CALL PATTERN(#2,108):: GOTO 830
810 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 820 ELSE CALL
PATTERN(#2,96):: RETURN
820 CALL PATTERN(#2,96):: IF GE=1 THEN R
ETURN ELSE GOTO 1090
830 FOR I=1 TO 50 :: NEXT I :: CALL COIN
C(ALL,CO):: IF CO=-1 THEN 840 ELSE CALL
PATTERN(#2,88):: RETURN
840 CALL PATTERN(#2,88):: IF GE=1 THEN R
ETURN ELSE GOTO 1090
850 IF GI=1 THEN 900 ELSE 860
860 CALL PATTERN(#2,112):: CALL MOTION(#
2,-20,20):: FOR I=1 TO 10 :: CALL COINC(
ALL,CO):: IF CO=-1 THEN GOTO 940 ELSE 87
0
870 NEXT I
880 CALL MOTION(#2,20,20):: FOR I=1 TO 1
0 :: CALL COINC(ALL,CO):: IF CO=-1 THEN
940 ELSE 890
890 NEXT I :: CALL PATTERN(#2,88):: CALL
 MOTION(#2,0,0):: CALL POSITION(#2,E,F):
: CALL LOCATE(#2,160,F):: RETURN
900 CALL PATTERN(#2,124):: CALL MOTION(#
2,-20,-20):: FOR I=1 TO 10 :: CALL COINC
(ALL,CO):: IF CO=-1 THEN GOTO 960 ELSE 9
10
910 NEXT I
920 CALL MOTION(#2,20,-20):: FOR I=1 TO
10 :: CALL COINC(ALL,CO):: IF CO=-1 THEN
```



„Cheng-Wang, ich freue mich ja wirklich, wenn Deine Brüder uns besuchen. Aber können sie nicht wie jeder andere an die Tür klopfen?“

```
 960 ELSE 930
930 NEXT I :: CALL PATTERN(#2,96):: CALL
 MOTION(#2,0,0):: CALL POSITION(#2,E,F):
: CALL LOCATE(#2,160,F):: RETURN
940 CALL SOUND(100,150,0,155,0,-2,0):: C
ALL MOTION(#2,0,0):: CALL POSITION(#2,E,
F):: CALL LOCATE(#2,160,F):: CALL PATTER
N(#2,88)
950 IF GE=1 THEN RETURN ELSE 1090
960 CALL SOUND(100,150,0,155,0,-2,0):: C
ALL MOTION(#2,0,0):: CALL POSITION(#2,E,
F):: CALL LOCATE(#2,160,F):: CALL PATTER
N(#2,96)
970 IF GE=1 THEN RETURN ELSE 1090
980 IF GI=2 THEN 990 ELSE 1010
990 GD=1 :: CALL PATTERN(#2,128):: CALL
KEY(0,K,S):: GOTO 370:
1000 GD=0 :: CALL PATTERN(#2,88):: CALL
KEY(0,K,S):: GOTO 370
1010 GD=2 :: CALL PATTERN(#2,136):: CALL
 KEY(0,K,S):: GOTO 370
1020 GD=0 :: CALL PATTERN(#2,96):: CALL
KEY(0,K,S):: GOTO 370
```



THAI BOXING — THE END

```
1030 GE=1 :: IF RI=2 THEN 1040 ELSE 1050
1040 CALL PATTERN(#1,128):: GOSUB 690 ::
 GE=0 :: CALL PATTERN(#1,88):: CALL KEY(
0,K,S):: GOTO 370
1050 CALL PATTERN(#1,136):: GOSUB 690 ::
 GE=0 :: CALL PATTERN(#1,96):: CALL KEY(
0,K,S):: GOTO 370
1060 P=P+1 :: IF P=5 THEN 1070 ELSE GOTO
 360
1070 P,W=0 :: IF GI=1 THEN CALL PATTERN(
#2,132)ELSE CALL PATTERN(#2,140)
1080 CALL SOUND(1300,-5,8):: FOR I=1 TO
100 :: NEXT I :: PU=PU+1 :: DISPLAY AT(4
,20)SIZE(3):PU :: IF PU=10 THEN GOTO 112
0 ELSE GOTO 340
1090 W=W+1 :: IF W=CW THEN 1100 ELSE CAL
L KEY(0,K,S):: GOTO 370
1100 W,P=0 :: IF RI=2 THEN CALL PATTERN(
#1,132)ELSE CALL PATTERN(#1,140)
1110 CALL SOUND(1300,-5,8):: FOR I=1 TO
100 :: NEXT I :: GU=GU+1 :: DISPLAY AT(5
,20)SIZE(3):GU :: IF GU=10 THEN GOTO 114
0 ELSE 340
1120 PU,GU=0 :: FOR I=1 TO 20 :: DISPLAY
 AT(4,8)SIZE(12):"" :: FOR VER=1 TO 20 :
: NEXT VER :: DISPLAY AT(4,8)SIZE(12):X$
 :: NEXT I
1130 GOSUB 1330 :: GOTO 1160
1140 GU,PU=0 :: FOR I=1 TO 20 :: DISPLAY
 AT(5,8)SIZE(12):"" :: FOR VER=1 TO 20 :
: NEXT VER :: DISPLAY AT(5,8)SIZE(12):Y$
 :: NEXT I
1150 FOR I=600 TO 300 STEP -30 :: CALL S
OUND(550,I,2,I-20,6,I+10,5):: NEXT I
1160 CALL DELSPRITE(ALL)
1170 DISPLAY AT(12,3)SIZE(20):"NEUES SPI
EL ? <J><N>" :: CALL KEY(0,K,S):: IF S=0
 THEN 1170 ELSE IF K=74 THEN 100 ELSE IF
 K=78 THEN 1190
1180 GOTO 1170
1190 CALL CLEAR :: END
1200 CALL CLEAR :: PRINT "THAI - BOXING"
1210 PRINT "ALS THAI-BOXER TRETEN SIE
GEGEN EINEN VON DREI WAEHL- BAREN GEGNER
N AN.WENN SIE"
1220 PRINT "IHREM GEGNER MEHRERE SCHLAE-
GE ODER TRITTE VERABREICHT  HABEN,DANN H
ABEN SIE IHN AUF"
1230 PRINT "DIE MATTE GELEGT UND BEKOMM-
EN  EINEN PUNKT.DAS GILT NA-TUERLICH AUC
H UMGEKEHRT.WER"
1240 PRINT "ZUERST 10 PUNKTE HAT,IST DER
SIEGER DES KAMPFES.          STEUERUNG DE
R SPIELFIGUR:"
1250 PRINT "S - NACH LINKS GEHEN
D - NACH RECHTS GEHEN          , - BOXEN"
1260 PRINT ". - TRETEN
L - SPRINGEN                   LEERTASTE -
ABWEHRSTELLUNG"
1270 PRINT "DRUECKEN SIE EINE TASTE !!!"
1280 CALL KEY(0,K,S):: IF S=0 THEN 1280
ELSE 100
1290 DATA 165,180,232,232,287,346,408,40
8,474,287,232,287,346,544,474
1300 RESTORE 1290 :: FOR I=1 TO 15 :: RE
AD Z
1310 CALL SOUND(+200,Z,0):: NEXT I :: RE
TURN
1320 DATA 233,246,261,246,277,261,277,29
3,311,293,277,293,311,261,246,277,261,29
3,311,329,349
1330 RESTORE 1320 :: FOR I=1 TO 21 :: RE
AD Z :: CALL SOUND(100,Z,0,349,5):: NEXT
 I :: RETURN
```

Letzte Meldung! Letzte Meldung!

ENTERPRISE hat die Preise für seine Super-Rechner gesenkt!

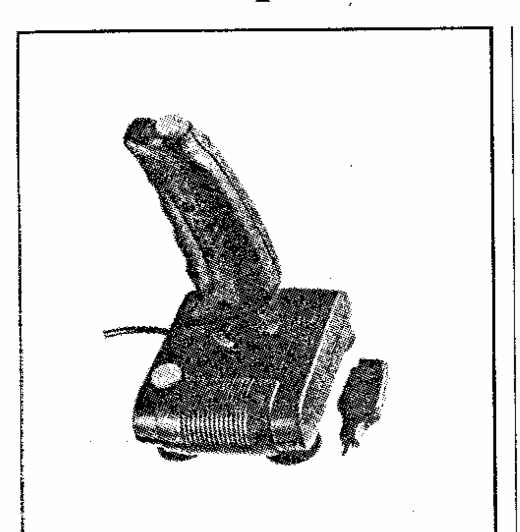
Der 64K-Computer kostet jetzt ganze 498 Mark, während der 128K-Enterprise schon für 798 Mark zu haben ist. Die Peripherie (ebenfalls im Preis drastisch gesenkt: Monitor, grün – 298,– / Monitor, farbig (Hi-Res) – 798,– / Drucker – 898,– / 3,5 Zoll Laufwerk mit Controller – 998,– / zweites Laufwerk ohne Controller – 698,– (720 KB-formatiert). Klasse Preise für einen Riesen-Computer!

Korrektur zum Programm „Brückenbau“ für den VC-20 COMPUTRONIC 1/2/86

```
8 C$(1)="$%$%$%$%$%$%$%":C$(2)="%$%$%$%$%$%$%$":G$="[illegible]
[illegible]"

49 IFPEEK(Z+26)=26ORPEEK(Z+26)=27ORPEEK(Z+27)=26THENPOKEZ,32:POKEZ+1,32:W=0:RETU
RN
```

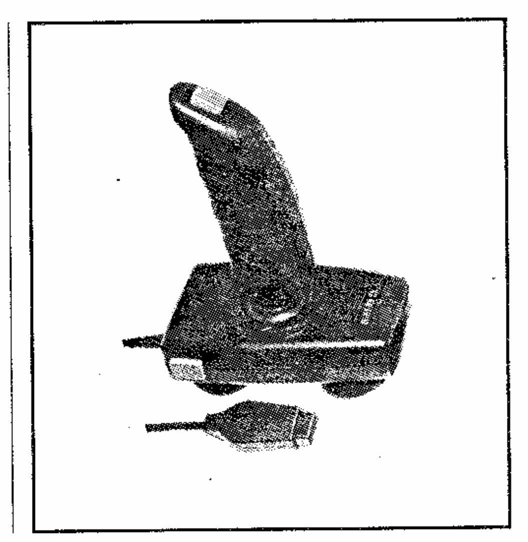
Ein Kurz-Tip für alle CPC-464-Besitzer – Joystickabfrage einmal anders!



Wie sicher allen Schneider-Usern bekannt ist befindet sich im Basic der Befehl JOY (nr.). Es existiert aber noch eine andere Möglichkeit den Joystick abzufragen, die in erster Linie für Maschinensprache-Programmierer interessant sein dürfte. Man braucht tatsächlich nur die Speicherstelle & B4F4 durch PEEK (in MC mit LD A (&&B4F%) abzufragen, um den Wert des Joysticks zu erhalten. Die gelieferte Zahl ist mit derjenigen identisch, die durch den JOY (nr)-Befehl geliefert wird.

(Ottfried Schmidt)





Auflösung und Gewinner unseres Wettbewerbes aus der Computronic-Ausgabe 1/2/86

Vorweg ein Dank an unsere Leser für ihr Engagement sprich rege Beteiligung. Bei der abgebildeten Persönlichkeit erhielten wir die kuriosesten Lösungsvorschläge. Doch handelte es sich weder um Roy Black, Roland Kaiser, Wolfgang von Goethe oder Louis den Sonnenkönig, sondern schlicht und einfach um William Shakespeare. Unter den zahlreichen Einsendungen haben wir 15 Gewinner ausgelost, die eine Kassette oder Diskette freier Wahl aus unseren Softwareangebot bald ihr Eigen nennen können:

Thomas Achterberg, Bad Camberg
(Atari – Super-Miner/Diamonds)
R. Dahnelt, Dreieich
(Commodore – Ufo/Skateboard-Sam)
Jörg Friedrich, Walldorf
(Commodore – Tron/Mercurios)
Hans Theo Funke, Bonn 2
(Atari – Groove)
Thomas Görlich, Darmstadt 12
(TI-99-Moon Race/Frogger/Slicks)
Jens Hellmann, Gravenwiesbach
(Atari – Donkey Kong/Kerzenheinz)
Peter Henke, Schwäbisch-Hall
(Schneider – Snider's Mace)
Michael Lotz, Essen
(VC-20 – Garten/Schloß Gruselstein)
Klaus Reiger, Saulgau 1
(ZX-Spectrum – Karl der Käfer)
Hans Joachim Reimann, Berlin 20
(Apple – Pro Data/Music Editor)
Gabriele Rößler, Dortmund 1
(Atari – Splitt)
Markus Rübsamen, Kriftel
(Apple – Pro Data/Music Editor)
Monika Krehl-Schröder, Bonn
(TI-99-Moon Race/Frogger/Slicks)
Manfred Statmann, Jade 1
(Apple – Wilder Westen/Karambolage)
Maskengenerator)

Atari-/Commodore-Joysticks am TI 99/4A

Da die Handlichkeit der Texas Joysticks für Spiele häufig ungenügend ist, hat sich der Autor mit der Pinbelegung auseinandergesetzt!

Der Quick Shot II

Diejenigen, die den Joystick „Quick Shot II" benutzen wollen, benötigen eine weitere Lötung im Joystick, um den „Auto fire" funktionsfähig zu bekommen.
Dazu werden die vier Kreuzschlitzschrauben an der Unterseite des Joystick herausgedreht und das Unterteil abgeklappt (Vorsicht, keine Kabel abreißen).
Nun muß eine Verbindung vom leichtgedrehten Haltestift des Schalters zu dem Pin mit dem roten Kabel an der Unterseite des Schalters gelötet werden.
Ohne diese Lötung funktioniert der Aktionsknopf bei Stellung des Auto fire auf „on" nicht.

Pin – Belegung bei TI 99/4A für Joysticks

Die Pinnummern wurden vom eingesteckten Stecker abgeschrieben.

Computereingang		Wert	Joyst. Atari/Commodore	
Pin	Bedeutung	ein / aus	Pin	Bedeutung
2	Ausgang Joyst. 2		1	– Höhenwert Ausgang
3	+ Höhenwert Eing.	+ 4 / 0	2	+ Höhenwert Ausgang
4	Keywert Eing.	+18 / –1	3	– Seitenwert Ausgang
5	– Seitenwert Eing.	– 4 / 0	4	– Seitenwert Ausgang
7	Ausgang Joyst. 1		6	Keywert Ausgang
8	– Höhenwert Eing.	– 4 / 0	8	Computer Eingang
9	+ Seitenwert Eing.	+ 4 / 0		

Anschließend wird das Unterteil wieder angeklappt und die vier Schrauben fest angedreht.
Für das Anschließen von zwei Joysticks benötigt man ein siebenadriges Flachbandkabel, zwei neunpolige Joystickstecker und eine neunpolige Joystickbuchse.
Das Kabel nicht zu kurz kaufen, lieber 10 cm zu lang als 1 cm zu kurz. 30 cm sollten aber reichen. Das Kabel kann auch, falls länger als 1 Meter als Verlängerung dienen.

(Jürgen Böhning)

— Tips und Tricks für den VC-20 —

A. ON ERROR GOTO ...

Folgende Einzeiler simuliert den Befehl unter Programmkontrolle:

```
A$="90"+stR(nnnnn)+cH(13):L=LEN
(A$):P0198,L:f0A=1TOL:P0630+A,ASC
(mI(A$,A)):nE
```

nnnnn ist die Zeilennummer der Errorbehandlungsroutine. Die etwas unorthodoxe Schreibweise der Befehle soll die Kurzformen darstellen; mit den Kurzformen umfaßt die Zeile insgesamt 77 Zeichen, es bleibt also noch Platz für eine fünfstellige Programmzeilennummer.

B. UNNEW.

```
10 FOR A=525 TO 577:READ d:POKE
   A,D:NEXT
20 POKE43,167:POKE44,2:POKE45,220:
   POKE46,2:CLR:SAVE "UNNEW",1,1
30 DATA 160,  3,200,177, 43,208,251,200
   200,152
40 DATA 160,  0,145, 43,165, 44,200,145,
   43,133
50 DATA  60,160,  0,132, 59,162,  0,200,
   208,  2
60 DATA 230, 60,177, 59,208,245,232,
   224,  3,208
70 DATA 242,200,208,  2,230, 60,132,
   45,164, 60
80 DATA 132, 46, 96
```

Der Einsatz erfolgt mit SYS 525:CLR
Das Programm ist an einem von dem Überschreiben halbwegs geschützten Platz im Stapelspeicher untergebracht und somit stets – auch nach einem Reset – verfügbar.

C. Verlangsamter PRINT

Folgende Zeile ermöglicht die Bestimmung der Druckgeschwindigkeit auf dem Bildschirm. Als Übergabeparameter ist ein vorher definierter String, hier A$, erforderlich. Die Geschwindigkeit wird durch die B-Schleife bestimmt:

```
FOR A = 1 TO LEN(A$):PRINT
            MID$(A$,A1);:
FOR B = 1 TO 22: NEXT B,A
```

D. Wieder: Tastaturverriegelung

Daß man die Tastatur mit POKE 649,0 verriegeln kann, ist längst bekannt: dieselbe Wirkung erzielt man auch mit:

```
POKE 37155,255
```

E. INSTRING

Folgende Zeile simuliert die INSTRING-Funktion: als Übergabeparameter müssen zwei String vorher definiert werden:
A$ ist der Suchstring
B$ ist der zu durchsuchende String
Die Zeile läßt sich als einzeiliges Unterprogramm abrufen, wenn die entsprechenden Befehle (s. Schreibweise!) in Kurzform eingegeben werden:

```
A=LEN(A$):B=LEN(B$):N=0:fOI=
1toB-A+1:IFA$()mI(B$mI,A)THENnEI:
N=I:I=B-A+1:nEI
```

In dieser Form bleibt noch Platz für eine fünfstellige Programmzeilennummer sowie reT [RETURN-Kuerzel]

F. Zentrierter Bildschirmausdruck

Als Übergabeparameter ist ein String für jede Bildschirmzeile erforderlich, z. B. hier A$:

```
PRINT TAB(( 22 - LEN(A$) ) / 2)A$
```

G. PRINTE Unterprogramm mit einem Parameter

Die inzwischen wohl weltbekannte Methode PRINTE [PRINT AT] zu simulieren:

```
POKE781,X:POKE782,Y:POKE783,0:
SYS65520:PRINT"string"
```

läßt sich zu einem Unterprogramm umwandeln, das nur gedruckt werden soll.

```
  10 P=255:GOSUB1000:PRINT"string"
1000 POKE781,p/22:POKE782,P-22*
     PEEK781:POKE783,0:
     SYS65520:RETURN
```

Wer nicht auf Anhieb den Wert für P ausrechnen kann, hat die Möglichkeit dieses dem Computer zu überlassen, und zwar mit der Zeile:

P = (Y*22)+X

H. Selbstmodifizierendes Programm

Wenn man die Speicherzellen kennt, in denen die tokenisierten BASIC Befehle stehen, kann man durch hineinPOKEn von anderen Tokenwerten den Programmablauf ändern, z. B.:

```
0 REM 200
```

Die Zeilen 1 bis 198 stehen für Programmphase 1 zur Verfügung

```
199 EQ=4+PEEK(43)+256*PEEK(44):
POKE EQ,137:END
```

Ab Zeile 200 läuft dann Programmphase 2. Durch Zeile 199 ist der Tokenwert für GOTO jetzt wirksam. Zeile 0 sieht beim Listing dann so aus:

```
0 GOTO 200
```

Dieses einfache Beispiel soll andere User anregen, mit den dadurch gebotenen Möglichkeiten zu experimentieren.

I. Tastaturbieper

Für diejenigen, die beim Eintippen eine akustische Bestätigung bevorzugen:

```
10 FOR A = 678 TO 701:READ D:POKE
   A,D:NEXT
20 DATA 169,  15,141,  14,144,120,169,
        78,141,  20
30 DATA   3,169,   3,141,  21,   3,  88,
        96,165,197
40 DATA 201,128,240,    7,101,197,105,
       128,141,  12
50 DATA 133,  76,191,234
```

Mit SYS 678 darf gebiept werden.

J. Blinkende Bildschirmzeile

Gelegentlich möchte man einen Ausdruck am Bildschirm von der Umgebung abheben: eine Möglichkeit besteht darin, den PRINT String an - und ausblinken zu lassen. Dies kann folgendermaßen geschehen:

```
  10 PO$="crsrhome+22*crsrdown"
  20 ME$="das,was blinken soll"
  30 Y= [bildschirmzeile]:X= [Spalte]
  40 NR= [anzahl der blinkphasen]:
     GOSUB1000
1000 BL=0
1020 FOR I= 1 TO NR
1030 PRINT LEFT$(PO$,Y+1);TAB(X);
     CHR$(18+ABS(128*BL));ME$
1040 TT=TI+J1
1050 BL=BL=0
1060 IF TI < TT THEN1070
1070 NEXT
1080 RETURN
```

TI in Zeile 1040 und 1060 ist die Systemvariable für die Systemuhr.

K. GOTO berechnet

Mit folgenden Einzeiler kann man zu einer vorher berechneten Zeile springen; Voraussetzung dafür ist, daß die Variable LN den Wert der Zeile enthält, bevor man zu dieser Zeile springt - P0785,188:P0786,200:POLN,PE(LN):LN=USR(0)



Der Kurs, der „ankommt"! **BASIC für Anfänger**



Basic-Kurs - Teil 5

Stringverarbeitung

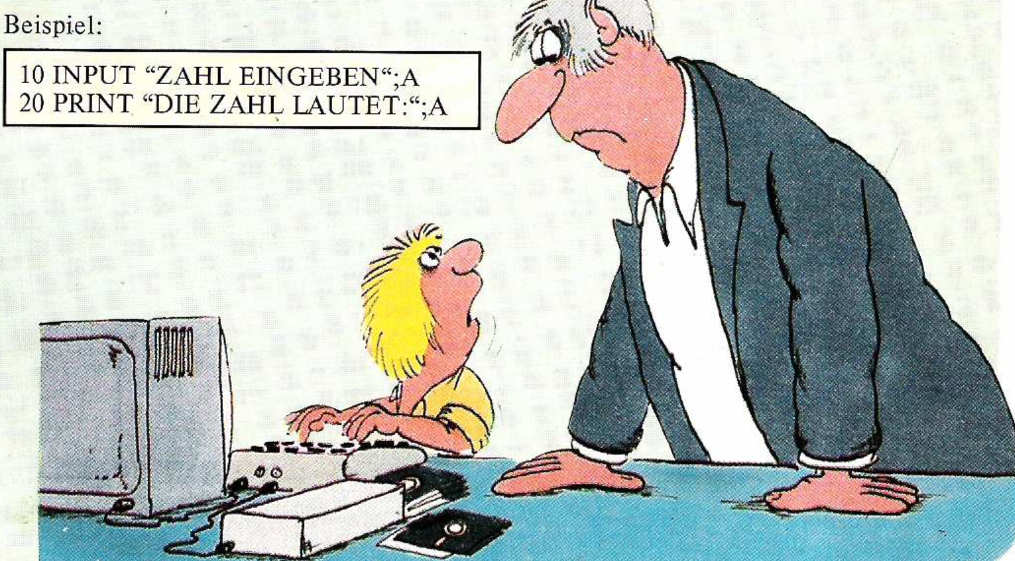
In den letzten beiden Teilen haben wir sehr viele neue Funktionen kennengelernt. Alle diese Funktionen haben gemeinsam, daß sie nur Zahlen verarbeiten können. Nun ist allgemein bekannt, daß Computer auf der Tastatur auch Buchstaben besitzen. Wir wollen uns nun mit den Funktionen befassen, die auch Buchstaben und andere Sonderzeichen verarbeiten können, man spricht in diesem Falle von Funktionen der Stringverarbeitung.

Was ist ein String?

Im ersten Teil unseres Kursus haben wir schon aufgezeigt, welche Zeichen der Computer darstellen und verarbeiten kann. Ein String ist eine Aneinanderreihung einer bestimmten Anzahl von beliebigen Zeichen. Ein Beispiel war unser kleines Mini-Programm im Kurs-Teil 2. Dort verwendeten wir einen String in einer PRINT- sowie in einer INPUT-Zeile.

Beispiel:

```
10 INPUT "ZAHL EINGEBEN";A
20 PRINT "DIE ZAHL LAUTET:";A
```



STRING ist das englische Wort für Kette und bedeutet in diesem Fall eine Art Zeichenkette.

Zum besseren Verständnis könnte man das Wort STRING durch das Wort Text, und das Wort Stringverarbeitung durch das Wort Textverarbeitung ersetzen.

Lesen Sie weiter auf der nächsten S.

Dieser Kurs läßt Sie zum perfekten Basic-Programmierer werden!

Von „Textverarbeitung" kann jedoch in dem obigen Beispiel noch keine Rede sein. Die Texte werden lediglich genauso, wie sie in der Programmanweisung vom Programmierer formuliert wurden, auch wieder auf dem Bildschirm ausgegeben (natürlich auch vorhandene Rechtschreibfehler).
Nun hat ein Computer natürlich noch etwas mehr zu bieten als INPUT und PRINT, als erstes wären hier die Stringvariablen zu nenen. Was man unter einer Variable versteht haben wir schon einmal angesprochen, jedoch kennen wir noch keine String - bzw. Textvariablen. Eine solche Variable unterscheidet sich von der numerischen Variable dadurch, daß dieser keine Zahl sondern ein ganzer String zugeordnet wird. Um dem Computer mitzuteilen wann es sich um eine numerische und wann es sich um eine Stringvariable handelt, muß der Variablenname entsprechend gekennzeichnet werden. Es ist üblich die Variablennamen, hinter denen sich Texte verbergen, mit einem angehängten $ (Dollar)-Zeichen von den Namen für numerische Variablen zu unterscheiden. Folgenden Textvariablen wären denkbar:

```
A$,B$,C$,T1$
```

Die meisten Computer erlauben auch längere Variablennamen wie:

```
PUNKTE$,OTTI$,NAME$
```

Denken Sie bei längeren Variablennamen daran, daß die einzelnen Computer eine unterschiedliche Anzahl von Zeichen interpretieren. Dadurch ist es möglich, daß die Variablennamen PUNKT$ und PUDEL$ als gleich interpretiert werden.
Wie bei den numerischen Variablen werden die Stringvariablen durch eine Zuweisung definiert. Im Gegensatz zu den numerischen Variablen kann hier der LET-Befehl entfallen.
Einige Beispiele:

```
10 A$= "OTTI GEHT NACH HAUS"
20 B1$="NAME:"
30 NAME$="FRANK"
40 PRINT A$
50 PRINT B1$
60 PRINT NAME$
```

Wir starten unser Programm:

```
RUN <RETURN>
OTTI GEHT NACH HAUS
NAME:
FRANK
```

Wie in dem Beispiel zu erkennen ist, muß bei der Zuweisung einer Stringvariable der eigentliche Text in Anführungszeichen gesetzt werden. Der Begriff Text kann recht locker gefaßt werden. Zu Recht spricht man nicht von „Buchstaben", sondern von „Zeichen". Dies soll verdeutlichen, daß man in einem String auch Sonderzeichen sowie Ziffern unterbringen darf:

```
10 A$="12345"
20 B$="$100 + $300"
30 C$="  "
```

Wie die Zeilen 20 und 30 unseres Beispieles zeigen, können auch Leerzeichen in einem String festgehalten werden. Viele Computer besitzen auch Grafikzeichen wie Figuren, Herzen oder Linien, auch solche Zeichen können in einem String verwendet werden. Es gibt jedoch einige Zeichen, die nicht verwendet werden können:

```
10 T$="ER SAGTE "HALLO" UND GING"
```

Die meisten Computer würden bei dieser Zeile annehmen, daß die Zeichenkette bei dem Wort „ER" beginnt und vor dem Wort „HALLO" endet. Dadurch können je nach Computertyp unterschiedliche Reaktionen hervorgerufen werden, entweder meldet sich der Computer mit einer Fehlermeldung oder er interpretiert die Zeile fehlerhaft und fährt im Programm weiter. Man sollte sich also immer erst im Handbuch seines Computers darüber informieren, wie dieser es gerne haben möchte. Im Zweifelsfalle sollte man durch eine kleine Testzeile (beispielsweise wie oben) feststellen wie der Computer auf das Anführungszeichen reagiert.

Wie lang darf ein String sein?

Wie wir in unseren Beispielzeilen schon gesehen haben, können in Stringvariablen Texte unterschiedlicher Länge zugewiesen werden. Die Programmiersprache Basic bietet dem User tatsächlich diese Möglichkeit. Je nach Computertyp und Basicversion ist eine maximale Stringlänge vorgegeben. Bevor Sie längere Strings verwenden, sollten Sie die Länge im Handbuch nochmals nachlesen.
Hier eine kleine Aufzählung einiger Computer:

Computertyp	Stringlänge
C64	255
VC20	255
C16	255
CPC 464	255
SPECTRUM	unbegrenzt
ZX81	unbegrenzt
MSX	100 (definierbar)

Die Stringlänge läßt sich leicht durch eine Basic-Zeile feststellen:

```
A$="":FOR I=1 TO 255:A$=A$+"A":NEXT I
```

Gibt Ihr Computer nach dem Start dieser Zeile keine Fehlermeldung aus, so kann Ihr Computer mindestens 255 Zeichen in einem String ablegen. Durch Erhöhen oder Erniedrigen der Zahl 255 können Sie schnell die maximale Stringlänge ermitteln.
Wird die maximale Stringlänge überschritten so melden sich die meisten Computer mit der Meldung "?STRING TOO LONG ERROR"
Einige Computer ignorieren auch einfach die überschüssigen Zeichen, ohne dies durch eine Fehlermeldung anzuzeigen. Dies stellt eine gefährliche Programmfalle dar und hat schon manchen zur Verzweiflung gebracht.

Handhabung von Strings

Ist einer Variable einmal ein String zugeordnet worden, so kann diese jederzeit auch wieder neu definiert werden:

```
10 A$="SCHMIDT"
20 PRINT A$
30 A$="MEIER"
40 PRINT A$
40 A$=""
60 PRINT A$
```

Wir starten unser Programm:

```
RUN <RETURN>
SCHMIDT
MEIER
```

In diesem Beispiel wird die Variable A$ drei mal definiert, wobei sie in Zeile 50 gelöscht wird.
Es gibt noch eine andere Methode, einer Textvariablen einen Wert zuzuweisen, und zwar mittels unserer INPUT-Anweisung. Folgendes Beispiel demonstriert dies:

```
10 REM *** STRING-ZUWEISUNG ***
20 INPUT A$
30 PRINT A$
40 GOTO 20
```

Ein anderes Beispiel:

```
10 REM ** STRING-ZUWEISUNG **
20 A$="NAME
30 B$="STRASSE"
40 C$="ORT"
50 INPUT A$,A1$
60 INPUT B$,B1$
70 INPUT C$,C1$
80 PRINT A$"="A1$
90 PRINT B$"="B1„
100 PRINT C$„=„C1$
```

Testen Sie selbst die Funktion dieser Programme und versuchen Sie ein eigenes kleines „STRING-Programm“ zu erarbeiten.
Im nächsten Teil gehen wir auf die ersten Stringfunktionen ein und schreiben ein kleines Programm.

Autor: Frank Brall

Literaturquellen:
Grundkurs in Basic, Ulrich Stroebel, Sybex-Verlag
Basic Brevier, Siegmar Wittig. Heinz Heise Verlag
Commodore Handbuch
Spectrum Handbuch
Commodore 116/16/plus4, Ekkehard Kaier, Vieweg Verlag

BASIC-KURS-TEIL 6 folgt in der nächsten Computronic

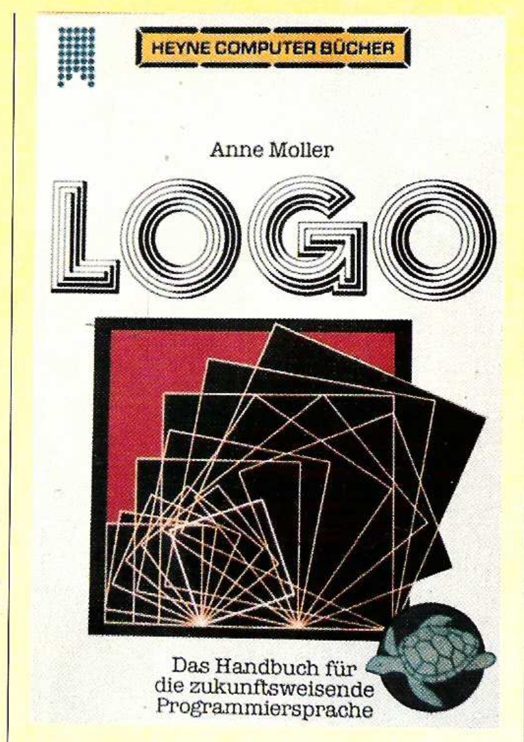
Computronic-Bücher-Kiste

Das Handbuch für die zukunftsweisende Programmiersprache

LOGO
Anne Moller

Logo, eine höhere Programmiersprache, wurde von Professor Seymour Papert in den Sechziger Jahren als Programmiersprache für Kinder erfunden. Nun, da diese Sprache auch in Deutschland auf dem Vormarsch ist, erschien bei Heyne zur rechten Zeit Anne Mollers Buch mit dem naheliegenden Titel „LOGO“.

Die Autorin vermittelt in ihrem Werk dem Leser einen umfassenden Einblick in Arbeitsweise und Möglichkeiten der Sprache Logo. Diese einfache und dennoch vielfältige Sprache hat es sich zur Aufgabe gestellt, Computer-Logik sichtbar zu machen. Mittels eines Lichtpunktes (der – bei einigen Computer-Systemen stilisierten – sogenannten „Schildkröte“) können auf dem Monitor durch einfache Wortbefehle grafische Gebilde gezeichnet werden. Darüberhinaus sollen Kinder mit dieser Sprache ein besseres Verständnis für Mathematik und Geometrie bekommen. Neben der Programmierung von beweglichen Bildern werden in „LOGO“ das gesamte Logo-Vokabular und seine Dialekte vorgestellt. Auch die Themengebiete „Logo im Unterricht“ und „Logo in der Informatik“ werden ausführlich erörtert.



Da die Programmiersprache Logo bereits in vielen Schulen angewandt wird, ist „LOGO“ von Anne Moller sicher ein nützlicher Ratgeber für fast jeden Computer-Anwender. Und noch ein Hinweis: Die Sprache Logo wird inzwischen von den meisten Heimcomputer-Herstellern angeboten; dazu gehören unter anderem IBM, Commodore, Atari, Tandy, Sinclair, Apple und viele andere.

„LOGO“ ist in „Heyne Computer-Buch“ (Wilhelm Heyne Verlag, München) erschienen. ISBN 3-453-47063-X, Preis 14,80 DM. (bez.)

Lernen mit dem Familien-Computer
Sabine Quinten-Eirich

„Lernen mit dem Familien-Computer“ richtet sich an Eltern und Erzieher auf der einen und an Kinder im Alter von fünf bis zwölf Jahren auf der anderen Seite (welche bei der Beschäftigung mit dem Buch jedoch auf die Unterstützung durch Erwachsene angewiesen sein werden).

Die Autorin, Sabine Quinten-Eirich, studierte Sozialpädagogik in München und nutzt und testet seit drei Jahren (als Mutter zweier Kinder sicher in sehr kompetenter Weise) Computer-Lehr- und -Lernprogramme. Neben einer umfangreichen Übersicht über das Angebot an Lehr- und Lernprogrammen gibt sie dem Leser die Möglichkeit, sich über Zeichen-, Musik- und Sprachlernprogramme für Kinder zu informieren.
Ihr Hauptaugenmerk legt Sabine Quinten-Eirich hierbei auf die Frage, ob überhaupt, wann und unter welchen Voraussetzungen Kinder an den Computer herangeführt werden sollten. Diese Problemstellung schlängelt sich gewissermaßen als „Roter Faden“ durch das gesamte Werk.

„Lernen mit dem Familien-Computer“ ist erschienen in der Reihe „Heyne Computer-Bücher“, Wilhelm Heyne Verlag, München. ISBN 3-453-47053-2. Preis 12,80 DM. (bez.)

Die Redaktion hat den neuen Sound-Champion des Monats gekürt!

Die Komposition „Flötenspiel" ist bestimmt nicht so schwungvoll und rhytmisch wie Kemal's Song, weist dafür aber andere entscheidende Vorzüge auf.
Diese dreistimmige Melodie wirkt sehr getragen und ist eindeutig von klassischen und barocken Elementen beeinflußt. Sascha Tayfeh ist es ohne Zweifel gelungen auf seinem C-64 einen täuschend realistischen Flötenspiel-Sound zu erzeugen. Für dieses Werk hat er sich die 250 Mark redlich verdient.
Das Programm arbeitet mit 3 Stimmen. Die erste Stimme stellt das Flötenspiel dar. Die zweiten und dritten Töne sind wechselnd.

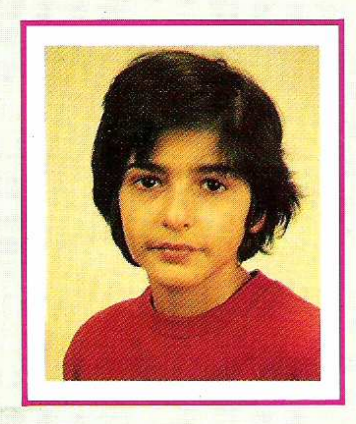
(Siegfried Görk)

Teilnahmebedingungen:

- Einsendung eines Datenträgers
- Beifügung von 1,40 DM in Briefmarken
- Kurze Beschreibung, Name des Titels
- Listing darf höchstens eine Heftseite lang sein!
- Kleines Autoren-Foto

Dies ist der würdige Nachfolger!

Unserem Aufruf aus der letzten Ausgabe (Januar/Februar) ist sehr bald eine positive Reaktion gefolgt. Wir haben mit Sascha Tayfeh und seiner barocken-klassischen Flötenkomposition einen würdigen Nachfolger für Kemal Ezcan, den Sound-Champion der letzten Ausgabe, gefunden. Saschas Komposition wurde übrigens auf einem C-64 geschrieben. Und nun an alle anderen potentiellen Computer-Komponisten! Laßt Eurer musikalischen Phantasie freien Lauf und deckt unsere Redaktion mit zahlreichen neuen Meisterwerken ein. Auf unseren „Musikus" des Monats warten wieder „satte" 250 Deutsche Mark.



Dieses Mal konnte sich Sascha Tayfeh (C-64) mit seinem „Flötenspiel" durchsetzen.

Der Soundtrack des Monats: FLÖTENSPIEL (C-64) von Sascha Tayfeh

```
10 PRINT"BITTE{SPACE}WARTEN.(RUN/STOP{SP
ACE}RESTORE=ENDE)":S=54272:FORL=STOS+24:
POKEL,0:NEXT                              <151>
20 DIMH(2,200),L(2,200),C(2,200):DIMFQ(1
1)                                        <19>
40 V(0)=17:V(1)=65:V(2)=33                <74>
50 POKES+10,3:POKES+22,240:POKES+23,244:
FORI=0TO11:READFQ(I):NEXT                 <208>
100 FORK=0TO2:I=0                         <177>
120 READNM                                <130>
130 IFNM=0THEN250                         <232>
140 WA=V(K):IFNM<0THENNM=-NM:WA=1         <20>
150 DR%=NM/128:OC%=(NMAND112)/16          <177>
160 NT=NMAND15:FR=FQ(NT):IFK=0THENOC%=OC
%+1                                       <30>
180 IFOC%=7THEN200                        <175>
190 FORJ=6TOOC%STEP-1:FR=FR/2:NEXT        <154>
200 HF%=FR/256:LF%=FRAND255               <196>
210 IFDR%=1THENH(K,I)=HF%:L(K,I)=LF%:C(K
,I)=WA:I=I+1:GOTO120                      <11>
220 FORJ=1TODR%-1:H(K,I)=HF%:L(K,I)=LF%:
C(K,I)=WA:I=I+1:NEXT                      <174>
230 H(K,I)=HF%:L(K,I)=LF%:C(K,I)=WA-1     <213>
240 I=I+1:GOTO120                         <90>
250 IFI>IMTHENIM=I                        <60>
260 NEXT                                  <135>
500 POKES+5,63:POKES+6,240:POKES+12,9:PO
KES+13,9:POKES+19,119:POKES+20,7          <170>
530 POKES+24,31                           <16>
550 FORI=0TOIM:POKES,L(0,I):POKES+7,L(1,
I):POKES+14,L(2,I)                        <26>
560 POKES+1,H(0,I):POKES+8,H(1,I):POKES+
15,H(2,I)                                 <11>
570 POKES+4,C(0,I):POKES+11,C(1,I):POKES
+18,C(2,I)                                <58>
580 FORT=1TO40:NEXT:NEXT:POKES+22,RND(1)
*64+190:POKES+10,RND(1)*14+1:GOTO500      <158>
600 DATA34334,36376,38539,40830           <163>
610 DATA43258,45830,48556,51443           <177>
620 DATA54502,57743,61176,64814           <225>
1000 DATA594,594,594,596,596              <15>
1010 DATA1618,587,592,587,585,331,336     <154>
1020 DATA1097,583,585,585,585,587,587     <157>
1030 DATA1609,585,331,337,594,594,593     <84>
1040 DATA1618,594,596,594,592,587         <61>
1050 DATA1616,587,585,331,336,841,327     <23>
1060 DATA1607                             <130>
1999 DATA0                                <185>
2000 DATA583,585,583,583,327,329          <222>
2010 DATA1611,583,585,578,578,578         <56>
2020 DATA196,198,583,326,578              <125>
2030 DATA326,327,329,327,329,326,578,583  <93>
2040 DATA1606,582,322,324,582,587         <192>
2050 DATA329,327,1606,583                 <189>
2060 DATA327,329,587,331,329              <252>
2070 DATA329,328,1609,578,834             <197>
2080 DATA324,322,327,585,1602             <171>
2999 DATA0                                <165>
3000 DATA567,566,567,304,306,308,310      <201>
3010 DATA1591,567,311,310,567             <93>
3020 DATA306,304,299,308                  <27>
3030 DATA304,171,176,306,291,551,306,308  <2>
3040 DATA310,308,310,306,295,297,299,304  <67>
3050 DATA1586,562,567,310,315,311         <23>
3060 DATA308,313,297                      <220>
3070 DATA1586,567,560,311,309             <144>
3080 DATA308,309,306,308                  <11>
3090 DATA1577,299,295,306,310,311,304     <86>
3100 DATA562,546,1575                     <183>
3999 DATA0                                <145>
```

Optimiertes Basicprogramm für den Profi

Fortsetzung von Seite 63

so kann man sich die ganze Sache natürlich vereinfachen, in dem man das neue Unterprogramm statt mit GOSUB mit GOTO aufruft. Das RETURN am Ende des zweiten Unterprogramms läßt das Programm zu dem Befehl zurückkehren, der dem Aufruf des ersten Unterprogramms folgt.

Beispiel:

```
10   GOSUB 1000
1000 REM Unterprogramm I
1010 A=SQR(25)
1020 GOSUB 2000
1030 RETURN
2000 REM Unterprogramm II
2010 PRINT A
2020 RETURN
```

Dieser Programmausschnitt ist gleichbedeutend mit folgendem:

```
10   GOSUB 1000
1000 REM Unterprogramm I
1010 A=SQR(25)
1020 GOTO 2000
2000 REM Unterprogramm II
2010 PRINT A
2020 RETURN
```

Das gleiche Verfahren läßt sich im Übrigen auch bei Assemblerprogrammen mit dem JSR und dem RTS-Befehl (beim 6502) anwenden. So kann Speicherplatz und Zeit gespart werden.

Fazit: Mit diesen Tips kann man in fast allen Fällen einiges an Rechenzeit einsparen. Probieren Sie es doch einmal selbst an Ihrem Rechner aus! – Die Mühe lohnt sich wirklich!!

Computerclub Husum C.C.H.

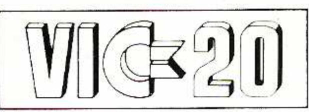
Husum (S.G.) – Der C.C.H. besteht seit Mai 1985. Ursprünglich war er als regionaler Club für Besitzer diverser Computer gedacht. Durch die Mailbox hat er sich aber über das gesamte Bundesgebiet und Österreich ausgebreitet. Die Altersstruktur der Mitglieder erstreckt sich von 15 bis 48 Jahren. Vorhandene Rechner sind zur Zeit C-64, Schneider, Atari, Acorn A und B.
Der Austausch von Erfahrungen und Informationen ist vorrangiges Ziel des Clubs. Statt des geplanten direkten Austausches erfolgt dieser wegen der großen Verbreitung im Rahmen eines noch unregelmäßig erscheinenden Infos und über die TINE-Mailbox (0 48 41 / 18 81), die 24 Stunden online ist.

Anfragen zum Club an:
Hans-Walter Latus
Postfach 13 47
2730 Zeven
Tel: 0 42 81 / 64 42
oder in der Mailbox an FESTUS.

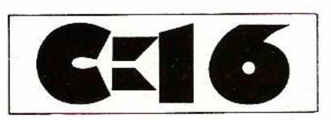
SOFTWARE-KATALOG ✱ COMPUTRONIC-SPITZEN-Software

C=64

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Mauern/Widerstand	8,--	15,--	C 41
Space-Comets/Erdspalte/Sprite-Data	15,--	23,50	C 51
Autostart/Bestellschein/Roadpainter	16,50	23,50	C 61
Hardcopy/Space-Fighter/ Data-Generator	15,50	19,50	C 71
Monster-Attack/Block-Painter/ Epson-Drucker	16,50	23,50	C 81
Projekt/Datenbank	16,--	23,50	C 91
Spiders/The Basic	16,50	23,50	C 101
High Noon/Skeet/Grafik/Designer	17,50	23,50	C 121
Painter/Star-Battle/Editor	17,50	23,50	C 22
Wüstenrallye/Jet-Pac/Black Moore Castle	17,50	23,50	C 32
Brieftaube/Cadelon	19,50	24,50	C 42
Ritter Erik/Grand Prix/Spritehilfe	19,50	24,50	C 52
Firebird/High Music	24,50	29,50	C 62
Moonsweeper/Scotti	24,50	29,50	C 72
Tron/Mercurious	24,50	29,50	C 13
Interceptor Base/Schotter	24,50	29,50	C 23



Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Bestellschein/Glücky	8,50	15,--	V 61
Multigraph/All Rammer	11,--	15,50	V 71
Zyklo/Meteorit	11,--	19,50	V 81
Garten/Schloß Gruselstein	14,--	19,50	V 91
Fressmann/Outlaw	14,--	19,50	V 101
Prost/Buffalo Bill	14,--	19,50	V 121
Joy Man/Powerpack	14,--	19,50	V 22
Der rasende Malocher	14,--	19,50	V 32
Matron/Obst	14,--	19,50	V 42
Race On/Cagy	14,--	19,50	V 52
Nürburg - 3 D/Düsi	14,--	19,50	V 62
Breaker/Expulsion	14,--	19,50	V 72
Brückenbau/Jango	16,--	21,50	V 13
Miner, der Fassadenstreicher/Inkaschatz	16,--	21,50	V 23

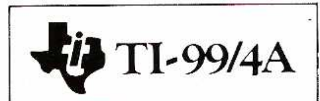


Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Horror Castle/Fantasy County	18,--	23,--	O 23

Bitte beachten Sie: Wenn Sie bei Ihrer Bestellung auf die sogenannte „Vorkasse" zurückgreifen, so ersparen Sie sich viel Geld. Das gilt besonders für unsere ausländischen Freunde; da die Nebenkosten derart hoch sind, schicken wir keine Bestellung per Nachnahme ins Ausland. Schikken Sie uns deshalb einen Bar- oder Verrechnungsscheck, Postanweisung oder Bargeld. (Die Nebenkosten in der BRD - Nachnahme - betragen 5 DM)

sinclair ZX81

Programm	Preis/ Kassette	Bestell-Nr.
Moon-Crash/ZX-Draw	10,--	Z 61
Tonprogramm/Aldebaran	10,--	Z 71
Reversi	10,--	Z 91
Panik Labyrinth	10,--	Z 101
Expedition	10,--	Z 121
Spinnen	14,50	Z 22
Spukhaus	14,50	Z 32
Frogger	14,50	Z 42
Olympiade	15,50	Z 52
Spider Mutants	15,50	Z 72
Der Elektroniker	10,--	Z 13



Programm	Preis/ Kassette	Bestell-Nr.
Drei-Kronen-Spiel/Zahlenputzen	8,50	T 41
Karl der Käfer/Alien-Landing	14,50	T 51
Jack the Digger I	14,50	T 61
Lift Bär/ASC II DEF Teil 1	14,--	T 71
Maya/ASC II DEF Teil 2	14,50	T 81
Transfer/Silverspar	14,50	T 91
Mother Duck/Screen Designer	14,50	T 101
Cave Man	8,--	T 121
Moon Race/Frogger/Slicks	19,50	T 22
Panzerschlacht	8,--	T 32
Maya II/Jagdszene Chikago	16,50	T 42
Raketen-Beschuß/Fire-Fox	16,50	T 52
Miner 99	14,50	T 62
Stardust	16,50	T 72
Jäger des verlorenen Schatzes	14,50	T 13
Thai-Boxing	12,50	T 23

ATARI 800 XL (*600)

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Mastermind*/Schlangenkrieg*	10,50	19,50	I 41
Tank-Battle/Oil Panic	12,50	-	I 51
Startup/Zeilen-Split/Tomstone-City	-	19,50	I 71
Painter/Hardcopy	14,--	19,50	I 81
The Big Quest/Fünf gewinnt	14,--	19,50	I 91
Splitt**	11,--	17,50	I 101
Ski/Mutation*	14,50	19,50	I 121
Super Miner/Diamonds	14,50	19,50	I 22
Donkey Kong/Kerzenheinz	16,50	19,50	I 32
Fighting/Escape from Earth	16,50	19,50	I 42
Höhlen-Herbert/Ball Harbour	16,50	19,50	I 52
Aquanaut*	16,50	19,50	I 62
Jumping Ghost/Soundtrack	16,50	19,50	I 72
Groove	-	29,50	I 13 A
Treter/Cavehunt	11,--	16,--	I 13 B
Top Tennis/Panzerschlacht	16,50	21,50	I 23

zu unglaublich günstigen Preisen ✳ ✳ ✳ SOFTWARE-KATALOG



Programm	Preis/ Diskette	Bestell-Nr.
Wilder Westen/Karambolage/Maskengenerator	19,50	A 41
Music-Maker/Mission: Adler Disk-Katalog	19,50	A 51
Snake/Super Datei/Shape-tables	19,50	A 61
Library/Fight	19,50	A 71
Reversal/Disk-Menue-Generator	19,50	A 81
Diamonds/Hilfsprogramm	19,50	A 91
Tic-Tac-Toe/Jumper	19,50	A 101
Donovan/Basic-Konverter	19,50	A 121
Funktionstasten/Painter/Bowling	19,50	A 22
Thunder/Castle of Doom	19,50	A 32
Hubschrauber	19,50	A 42
Widerstandsdecodierung/Soft-Talker	16,50	A 52
Atlantic Gold/Vier gewinnt	16,50	A 72
Prodata/Music Editor	19,50	A 13
Pac Boy	16,50	A 23



Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Super Miner	14,--	24,--	SR 41
CPC-Bert	14,--	24,--	SR 52
Conan's Castle	25,50	15,50	SR 62
Snider's Mace	16,50	26,50	SR 72
Schneider Panik/Killer Ship	18,50	28,50	SR 13
Midnight/Horror Caves	18,50	28,50	SR 23

sinclair Spectrum

Programm	Preis/ Kassette	Bestell-Nr.
Inventur	12,--	S 41
Missle-Comment	8,50	S 51
Defender Lui der Wurm Alternativer Zeichensatz	13,50	S 61
Matheprogramm Bongo-Beecatcher	12,50	S 71
Solitaire Superstat Kleinstes gem Vielfache	14,50	S 81
Jump about	14,50	S 91
Pac-Man Oil Panic	14,50	S 101
Frogger	16,--	S 121
Jump	14,50	S 22
Jet Set Freddie	8,--	S 32
Andromeda	14,50	S 42
Totenkopf	14,50	S 52
Willibald	15,50	S 62
Squares	15,50	S 72
Karl der Käfer	15,50	S 13
Otto Schweinsohr/Ausschnittkopierer	16,50	S 23

Machen Sie es wie Bodo Ballermann*:

Sichern auch SIE sich rechtzeitig die superpreiswerten SOFTWARE-Angebote von TRONIC! Bei diesen tollen Preisen gibt's nur eins:



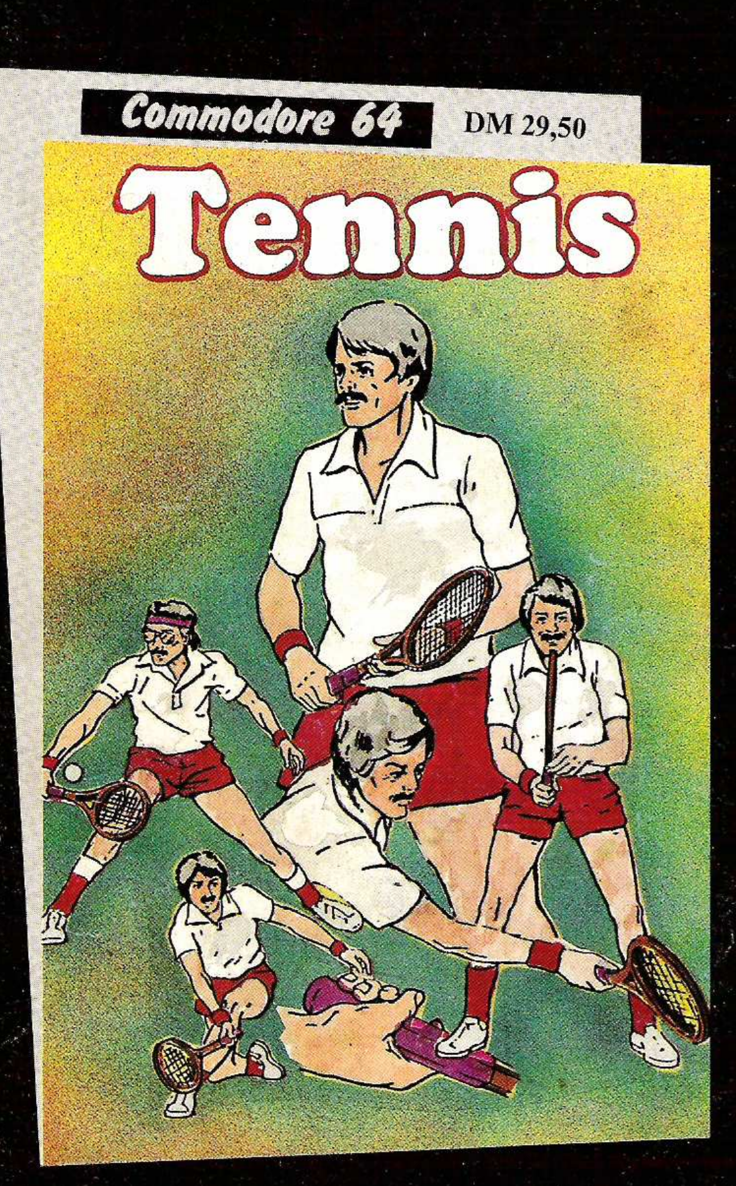
* (Außenstürmer des FC Freakdorf, 21 A-Länderspiele)



„Na komm schon, Bodo, laß mich doch wenigstens mal in Deine Tasche reinschauen!"



TRE GAMES



Das TENNIS-Fieber hat uns alle ergriffen. Nicht zuletzt durch die großartigen Erfolge von Boris Becker ist der „Weiße Sport" zu einem Volkssport geworden! In unserem Game treten zwei Tennis-Cracks gegeneinander an; sie schenken sich nichts; um jeden Ball wird gekämpft. Das Spiel basiert auf den Grundregeln des internationalen Tennis'. Hart umkämpfte Tie-Breaks sind ebenso an der Tagesordnung wie ein zermürbendes Fünfsatz-Spiel.
Sie haben hierbei die Möglichkeit, Ballgeschwindigkeit und Ballwinkel zu variieren. Aufgeschlagen wird per „Knopfdruck". Ein Superspiel in 3 D-Effekt.



Der alte Magier will einen Zaubertrank brauen. Dazu benötigt er allerlei Zutaten.
Also ist er unterwegs, um Spinnen, Kraken, Pilze und andere Gegenstände einzusammeln. Doch Hexen, Fledermäuse und anderes Ungetier versuchen ihn daran zu hindern. Der Magier ist aber nicht hilflos!
Die Monster kann er mit Zaubersternen abwehren. Hat er eine Zutat gefunden, so muß er sie in den Topf werfen. Ein Spaß für die ganze Familie.





Postfach · D-3444 Wehretal 1 · Tel. 05651/40643